



[Alauda DevOps Connectors Docs](#) > [API Reference](#) > [Kubernetes APIs](#) > Operator APIs

Operator APIs

ConnectorsCore

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsCore is the Schema for the connectorscore API.

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsCoreSpec defines the desired state of ConnectorsCore.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsCoreSpec defines the desired state of ConnectorsCore.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template	object	Template describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.spec.workloads[].template

Description

Template describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

Type

object

Property	Type	Description
metadata	ObjectMeta ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
spec	object	Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status ↗

.spec.workloads[].template.spec

Description

Specification of the desired behavior of the pod. More info:

<https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

Type

object

Required

containers

Property	Type	Description
activeDeadlineSeconds	integer	Optional duration in seconds the pod may StartTime before the system will actively tr containers. Value must be a positive intege

Property	Type	Description
affinity	object	If specified, the pod's scheduling constrain
automountServiceAccountToken	boolean	AutomountServiceAccountToken indicates be automatically mounted.
containers	array	List of containers belonging to the pod. Co removed. There must be at least one conta
dnsConfig	object	Specifies the DNS parameters of a pod. Pa to the generated DNS configuration based
dnsPolicy	string	Set DNS policy for the pod. Defaults to "Cl 'ClusterFirstWithHostNet', 'ClusterFirst', 'De in DNSConfig will be merged with the polic DNS options set along with hostNetwork, y to 'ClusterFirstWithHostNet'.
enableServiceLinks	boolean	EnableServiceLinks indicates whether info injected into pod's environment variables, r Optional: Defaults to true.

Property	Type	Description
<code>ephemeralContainers</code>	<code>array</code>	List of ephemeral containers run in this pod. Ephemeral containers complete their life cycle within an existing pod to perform user-initiated actions that require elevated privileges that the pod cannot be specified when creating a pod, and cannot be added to the pod spec. In order to add an ephemeral container to an existing pod's ephemeralcontainers subresource.
<code>hostAliases</code>	<code>array</code>	HostAliases is an optional list of hosts and their corresponding IP addresses. Hosts must be present in the /etc/hosts file if specified.
<code>hostIPC</code>	<code>boolean</code>	Use the host's ipc namespace. Optional: Default to false
<code>hostNetwork</code>	<code>boolean</code>	Host networking requested for this pod. Use the host's network stack. Optional: Default to false. If the option is set, the ports that will be used must be requested from the host's network namespace.
<code>hostPID</code>	<code>boolean</code>	Use the host's pid namespace. Optional: Default to false
<code>hostUsers</code>	<code>boolean</code>	Use the host's user namespace. Optional: Default to false. If the option is set, the pod will be run in the host's user namespace. This requires a kernel module with CAP_SYS_MODULE. Only available on kernel version >= 4.11.0. If the option is not set, the container will be run in the container's user namespace. Setting false is useful if the container needs to have root privileges on the host. This field is only applicable to containers created for the pod. Setting false is useful if the container needs to have root privileges on the host. This field is only applicable to containers created for the pod.

Property	Type	Description
hostname	string	Specifies the hostname of the Pod If not set to a system-defined value.
imagePullSecrets	array	ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images listed in the podSpec. More info: https://kubernetes.io/docs/concepts/configuration/secret#imagepullsecrets-on-a-pod
initContainers	array	List of initialization containers belonging to the pod. Init containers are designed to perform user-initiated tasks before the 'main' container(s) in the pod are started. If an init container fails, the pod is considered to have failed and is handled according to the pod's restartPolicy. In this case, the pod may be rescheduled to another host and the initialization process may start from the first container. Init containers may not have Lifecycle actions, Startup probes, and Readiness probes, but may utilize liveness probes. The resourceRequirements field is optional for init containers, but the pod's resourceRequirements account during scheduling by finding the highest resource requirement across all containers in a pod, and then using the max of that value. Limits are applied to init containers in a pod. Init containers are added or removed. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/init-containers/
nodeName	string	nodeName indicates in which node this pod is to be scheduled. If the field is set, the kubelet for this node becomes a candidate for scheduling the pod. This field should not be used to express a preference for a specific node. https://kubernetes.io/docs/concepts/scheduling/kubernetes-scheduler/#pod-node/#nodename

Property	Type	Description
nodeSelector	object	NodeSelector is a selector which must be 1 which must match a node's labels for the p info: https://kubernetes.io/docs/concepts/ci
os	object	Specifies the OS of the containers in the p restricted if this is set. If the OS field is set to linux, the following f securityContext.windowsOptions If the OS field is set to windows, following 1 • spec.hostPID • spec.hostIPC • spec.hostUsers • spec.securityContext.appArmorProfile • spec.securityContext.seLinuxOptions • spec.securityContext.seccompProfile • spec.securityContext.fsGroup • spec.securityContext.fsGroupChangeP • spec.securityContext.sysctls • spec.shareProcessNamespace • spec.securityContext.runAsUser • spec.securityContext.runAsGroup • spec.securityContext.supplementalGroi • spec.securityContext.supplementalGroi • spec.containers[*].securityContext.app/ • spec.containers[*].securityContext.seLii • spec.containers[*].securityContext.secc

Property	Type	Description
		- spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readonlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsNonRoot - spec.containers[*].securityContext.runAsUser
overhead	object	Overhead represents the resource overhead of a Pod given RuntimeClass. This field will be auto-populated by the RuntimeClass admission controller. If the FeatureGate is enabled, overhead must not be set in Pod admission controller will reject Pod creation if already set. If RuntimeClass is configured for the Pod, overhead will be set to the value defined in the corresponding RuntimeClass. If not set, overhead will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/482-pod-overhead
preemptionPolicy	string	PreemptionPolicy is the Policy for preempting existing pods. Never, DoNotPreempt, PreemptLowerPriority. Defaults to Never.
priority	integer	The priority value. Various system components use this field to find the priority of the pod. When Priority Admission Control is enabled, setting this field is required. The admission controller will reject a pod without a priority class name. The higher the value, the higher the priority.
priorityClassName	string	If specified, indicates the pod's priority. "system-cluster-critical" and "system-node-critical" are two special keywords which indicate the highest priorities within the cluster.

Property	Type	Description
		former being the highest priority. Any other PriorityClass object with that name. If not set or zero if there is no default.
readinessGates	array	If specified, all readiness gates will be evaluated when all its containers are ready AND all readiness gates have status equal to "True" More info: https://git.k8s.io/enhancements/keps/sig-network/0000-readiness-gates
resourceClaims	array	ResourceClaims defines which ResourceClaims the Pod needs before the Pod is allowed to start. The resourceClaims list contains the names of the ResourceClaims which consume them by name. This is an alpha field and requires enabling the ResourceClaims feature gate. This field is immutable.
restartPolicy	string	Restart policy for all containers within the pod. Some contexts, only a subset of those values are allowed. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle/#restart-policy
runtimeClassName	string	RuntimeClassName refers to a RuntimeClass object in the Kubelet config which should be used to run this pod. If no class is named, the pod will not be run. If unspecified, the default runtime handler will be used, which is an implicit class with the name "default". More info: https://kubernetes.io/docs/concepts/workloads/pods/runtime-handlers/

Property	Type	Description
<code>schedulerName</code>	<code>string</code>	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
<code>schedulingGates</code>	<code>array</code>	SchedulingGates is an opaque list of values that must all be satisfied in order to schedule the pod. If schedulingGates is not empty, the pod will be scheduled by the scheduler. SchedulingGates can only be set at pod creation time and the scheduler will not attempt to modify it afterwards.
<code>securityContext</code>	<code>object</code>	SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See the documentation for individual fields for details.
<code>serviceAccount</code>	<code>string</code>	DeprecatedServiceAccount is a deprecated service account to use to run this pod. Deprecated: Use serviceAccountName instead.
<code>serviceAccountName</code>	<code>string</code>	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

Property	Type	Description
<code>setHostnameAsFQDN</code>	<code>boolean</code>	If true the pod's hostname will be configured as the leaf name (the default). In Linux containers this means setting the registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Fqdn to FQDN. If a pod does not have FQDN, then the pod will not have a domainname at all.
<code>shareProcessNamespace</code>	<code>boolean</code>	Share a single process namespace between all containers in this pod. When this is set containers will be able to view and signal processes in other containers in the same pod, and the first process will have PID 1. HostPID and ShareProcessNamespace cannot both be set. Optional: Default to false.
<code>subdomain</code>	<code>string</code>	If specified, the fully qualified Pod hostname will be <code><pod-name>.<subdomain>.</code> pod will not have a domainname at all.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no graceful termination is required). If the value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are expected to cleanly stop. When grace period is 0, the pod needs to terminate gracefully. If the value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are expected to cleanly stop.
<code>tolerations</code>	<code>array</code>	If specified, the pod's tolerations.

Property	Type	Description
<code>topologySpreadConstraints</code>	<code>array</code>	TopologySpreadConstraints describes how across topology domains. Scheduler will spread the constraints. All topologySpreadConstra
<code>volumes</code>	<code>array</code>	List of volumes that can be mounted by co info: https://kubernetes.io/docs/concepts/st

`.spec.workloads[].template.spec.affinity`

Description

If specified, the pod's scheduling constraints

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	<code>object</code>	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	<code>object</code>	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

`.spec.workloads[].template.spec.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; then choose the node(s) with the highest sum.

Property	Type	Description
		sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirement specified by this field is not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement specified by this field ceases to be met at some point during pod execution (e.g. due to update), the system may or may not try to eventually evict the pod from its node.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, `requiredDuringScheduling` affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding `matchExpressions`; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
<code>preference</code>	<code>object</code>	A node selector term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

e.matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.

Property	Type	Description
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
<code>nodeSelectorTerms</code>	array	Required. A list of node selector terms. The terms are ORed.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelect`

orTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector

orTerms[].matchFields[].values[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding the weight of each element that is matched by the node. Elements that are not matched by the node will not be counted, but they will still contribute to the sum if they have a non-zero weight. This field may be used to give higher priority to scheduled pods in certain cases.

Property	Type	Description
		adding "weight" to the sum if the node has p which matches the corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirem specified by this field not met at scheduling time, the pod will not scheduled onto the n If the affinity requirem specified by this field cease to be met at sc point during pod execution (e.g. due to pod label update), the system may or may r try to eventually evict pod from its node. Wl there are multiple elements, the lists of nodes corresponding each podAffinityTerm intersected, i.e. all ter must be satisfied.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod

Property	Type	Description
		labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

`yTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

`yTerm.namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces`

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector` or

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchL

abelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace`

`eSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expression specified by this field, it may choose a node that violates one or more the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution anti-affinity expression etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has p

Property	Type	Description
		which matches the corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at the current scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm intersected, i.e. all terms must be satisfied.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

```
.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm
```

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

finityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values
[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

finityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismat

chLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

paceSelector.matchExpressions[].values[]

Type

string

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paceSelector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces**

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces[]**

Type

string

`.spec.workloads[].template.spec.containers`

Description

List of containers belonging to the pod. Containers cannot currently be added or removed.
There must be at least one container in a Pod. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[]`

Description

A single application container that you want to run within a pod.

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image if this is not provided. Variable references <code>\$(VAR)</code> expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code> , which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code> . Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code> , which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code> . Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must match the container's environment.

Property	Type	Description
		C_IDENTIFIER. All invalid keys will be reported when the container is starting. When a key exists in multiple sources, the value associated with the last source takes precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images field is optional to allow higher level config management to default or override container images in workload Deployments and StatefulSets.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images
lifecycle	object	Actions that the management system should take in response to container lifecycle events. Cannot be updated.
livenessProbe	object	Periodic probe of container liveness. Container v1 will fail if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL).

Property	Type	Description
		Cannot be updated.
ports	array	List of ports to expose from the container. Not specifying here DOES NOT prevent that port from being exposed, which is listening on the default "0.0.0.0" address. If the container will be accessible from the network. Modifying an array with strategic merge patch may corrupt the information. See https://github.com/kubernetes/kubernetes/issues/20380 Cannot be updated.
readinessProbe	object	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
resizePolicy	array	Resources resize policy for the container.
resources	object	Compute Resources required by this container. Cannot be updated. More info: https://kubernetes.io/docs/concepts/configuration/resource-limits
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers, this field is not specified, the restart behavior is controlled by the pod's restartPolicy.

Property	Type	Description
		Pod's restart policy and the container type. Setting RestartPolicy as "Always" for the init container will have the following effect: this init container will be continuously restarted upon exit until all regular containers have terminated. Once all regular containers have completed, all init containers with RestartPolicy "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar". Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next container starts immediately after this init container completes, after any startupProbe has successfully completed.
securityContext	object	SecurityContext defines the security options the container will be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/
startupProbe	object	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully completes. If this probe fails, the Pod will be restarted. This can be used to probe parameters at the beginning of a Pod's life cycle. The probe might take a long time to load data or warm a cache before entering steady-state operation. This cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
stdin	boolean	Whether this container should allocate a buffer for stdin data at container runtime. If this is not set, reads from stdin will return EOF.

Property	Type	Description
		container will always result in EOF. Default is false
stdinOnce	boolean	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When false, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process from stdin will never receive an EOF. Default is false
terminationMessagePath	string	Optional: Path at which the file to which the container termination message will be written is mounted in the container's filesystem. Message written is intended to be of final status, such as an assertion failure message. Message length across all containers will be limited by the node if greater than 4096 bytes. Defaults to /dev/termination-log. Cannot be updated.
terminationMessagePolicy	string	Indicate how the termination message should be handled. If set to File, will use the contents of terminationMessagePath to output the container status message on both success and failure. If set to FallbackToLogsOnError, FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 2 lines, whichever is smaller. Defaults to File. Cannot be updated.

Property	Type	Description
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be u container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystems updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be co container image. Cannot be updated.

`.spec.workloads[].template.spec.containers[].args`

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$ (VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

`array`

.spec.workloads[].template.spec.containers[].args[]

Type

string

.spec.workloads[].template.spec.containers[].command

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.containers[].command[]

Type

string

.spec.workloads[].template.spec.containers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

.spec.workloads[].template.spec.containers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.containers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key of a ConfigMap.
fieldRef	object	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
resourceFieldRef	object	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
secretKeyRef	object	Selects a key of a secret in the pod's namespace

.spec.workloads[].template.spec.containers[].env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

object

Required

key

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap or its key must be defined

.spec.workloads[].template.spec.containers[].env[].valueFrom.fieldRef

Description

Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Type

object

Required

fieldPath

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.containers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>string</code>	Specifies the output format of the exposed resources, defaults to "1"

Property	Type	Description
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.containers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key of the secret to select from. Must be a valid secret key.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.containers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[].envFrom[]`

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	<code>object</code>	The ConfigMap to select from
<code>prefix</code>	<code>string</code>	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	<code>object</code>	The Secret to select from

`.spec.workloads[].template.spec.containers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap must be defined

.spec.workloads[].template.spec.containers[].envFrom[].secretRef

Description

The Secret to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.containers[].lifecycle`

Description

Actions that the management system should take in response to container lifecycle events.
Cannot be updated.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
<code>preStop</code>	<code>object</code>	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the

Property	Type	Description
		container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗

`.spec.workloads[].template.spec.containers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec.command[]

Type

string

.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name``value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perform.
<code>sleep</code>	<code>object</code>	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	<code>object</code>	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.tcpSocket`

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].livenessProbe`

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies an action involving a GRP
<code>httpGet</code>	object	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	integer	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
<code>periodSeconds</code>	integer	How often (in seconds) to perform the prot 10 seconds. Minimum value is 1.
<code>successThreshold</code>	integer	Minimum consecutive successes for the pr considered successful after having failed. I

Property	Type	Description
		Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully terminate upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via a kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].livenessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.workloads[].template.spec.containers[].livenessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.workloads[].template.spec.containers[].livenessProbe.exec.command[]

Type

string

`.spec.workloads[].template.spec.containers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ^). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.workloads[].template.spec.containers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].livenessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].ports`

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.containers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.containers[].readinessProbe`

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period duration in seconds after the processes running on the pod are sent a termination signal and the time until the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].readinessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.containers[].readinessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.workloads[].template.spec.containers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>	<code>int32</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.containers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.containers[].resources`

Description

Compute Resources required by this container. Cannot be updated. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.containers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.containers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.containers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.containers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.containers[].securityContext`

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.containers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.containers[].securityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.containers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.containers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.containers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.containers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running on the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.containers[].startupProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.containers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.containers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.containers[].startupProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.containers[].volumeDevices[]`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.containers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.containers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath` `name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain '..'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.dnsConfig`

Description

Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy.

Type

`object`

Property	Type	Description
<code>nameservers</code>	<code>array</code>	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
<code>options</code>	<code>array</code>	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.
<code>searches</code>	<code>array</code>	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

`.spec.workloads[].template.spec.dnsConfig.nameservers`

Description

A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.

Type

`array`

`.spec.workloads[].template.spec.dnsConfig.nameservers[]`

Type

`string`

`.spec.workloads[].template.spec.dnsConfig.options`

Description

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type

array

`.spec.workloads[].template.spec.dnsConfig.options[]`

Description

PodDNSConfigOption defines DNS resolver options of a pod.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Required.
<code>value</code>	<code>string</code>	

`.spec.workloads[].template.spec.dnsConfig.searches`

Description

A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

Type

array

`.spec.workloads[].template.spec.dnsConfig.searches[]`

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers`

Description

List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[]`

Description

An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.

Type

`object`

Required

`name`

Property	Type	Description
<code>args</code>	<code>array</code>	Arguments to the entrypoint. The image's CMD is not provided. Variable references \$(VAR_NAME) using the container's environment. If a variable c

Property	Type	Description
		resolved, the reference in the input string will be Double \$\$ are reduced to a single \$, which allow the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" string literal "\$(VAR_NAME)". Escaped reference expanded, regardless of whether the variable ex Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
command	array	Entrypoint array. Not executed within a shell. The ENTRYPOINT is used if this is not provided. Vari \$(VAR_NAME) are expanded using the containe If a variable cannot be resolved, the reference in will be unchanged. Double \$\$ are reduced to a s allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ Escaped references will never be expanded, reg whether the variable exists or not. Cannot be up https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
env	array	List of environment variables to set in the contain updated.

Property	Type	Description
<code>envFrom</code>	<code>array</code>	List of sources to populate environment variables in the container. The keys defined within a source must match those of the container. C_IDENTIFIER. All invalid keys will be reported as an error when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
<code>image</code>	<code>string</code>	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images
<code>imagePullPolicy</code>	<code>string</code>	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images
<code>lifecycle</code>	<code>object</code>	Lifecycle is not allowed for ephemeral containers
<code>livenessProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>name</code>	<code>string</code>	Name of the ephemeral container specified as a pod's spec. This name must be unique among all containers, and ephemeral containers.

Property	Type	Description
<code>ports</code>	<code>array</code>	Ports are not allowed for ephemeral containers.
<code>readinessProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	Resources are not allowed for ephemeral containers use spare resources already allocated
<code>restartPolicy</code>	<code>string</code>	Restart policy for the container to manage the restart of each container within a pod. This may only be set on non-ephemeral containers. You cannot set this field on ephemeral containers.
<code>securityContext</code>	<code>object</code>	Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the SecurityContext override the equivalent fields of PodSecurityContext.
<code>startupProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data at container runtime. If this is not set, reads from stdin will always result in EOF. Default is false

Property	Type	Description
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When false, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on pod start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the pod is restarted. If this flag is false, a container process from stdin will never receive an EOF. Default is false
<code>targetContainerName</code>	<code>string</code>	If set, the name of the container from PodSpec to which the ephemeral container targets. The ephemeral container will then run in the namespaces (IPC, PID, etc) of this container. If not set, then the ephemeral container uses the namespaces of the Pod spec. The container runtime must implement support for namespace targeting. If the runtime does not support namespace targeting, the result of setting this field is undefined.
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container's termination message will be written is mounted in the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. The total message length across all containers will be limited by the node. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. If set to truncate, will use the contents of terminationMessagePath and truncate it to the specified length.

Property	Type	Description
		container status message on both success and failure. FallbackToLogsOnError will use the last chunk of output if the termination message file is empty and exited with an error. The log output is limited to 2048 lines, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for stdin. Requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used within the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the container runtime's default will be used, which might be configurable. Cannot be updated.

`.spec.workloads[].template.spec.ephemeralContainers[].args`

Description

Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `$(foo $(foo $(bar)))`.

"\$\$ (VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].args[]

Type

string

.spec.workloads[].template.spec.ephemeralContainers[].command

Description

Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].command[]

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. Must be a C_IDENTIFIER.
<code>value</code>	<code>string</code>	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME)

Property	Type	Description
		syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

object

Required

resource

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>		Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[]`

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	The ConfigMap to select from
prefix	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
secretRef	object	The Secret to select from

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[].secretRef`

Description

The Secret to select from

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle`

Description

Lifecycle is not allowed for ephemeral containers.

Type

object

Property	Type	Description
<code>postStart</code>	object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks
<code>preStop</code>	object	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].ifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].ifecycle.postStart.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be

Property	Type	Description
		an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility.

Property	Type	Description
		There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec.command[]

Type

string

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.

Property	Type	Description
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.tcpSocket`

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when

tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimum
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period in seconds after the processes run

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u:
timeoutSeconds	integer	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. <a href="https://kubernetes.io/docs/concepts/worklo
lifecycle#container-probes">https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`

Required

`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.ephemeralContainers[].li venessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].ports`

Description

Ports are not allowed for ephemeral containers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe`

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-life-cycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. Defaults to 30 seconds. <code>spec.terminationGracePeriodSeconds</code> is used if not set.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/probe-lifecycle#container-probes

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.ephemeralContainers[].resources`

Description

Resources are not allowed for ephemeral containers. Ephemeral containers use spare resources already allocated to the pod.

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.ephemeralContainers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.ephemeralContainers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.ephemeralContainers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext`

Description

Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe`

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.
failureThreshold	integer	Minimum consecutive failures for the probe considered failed after having succeeded.

Property	Type	Description
		Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRP
httpGet	object	HTTPGet specifies the http request to perf
initialDelaySeconds	integer	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
periodSeconds	integer	How often (in seconds) to perform the prot 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
tcpSocket	object	TCPSocket specifies an action involving a
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim

Property	Type	Description
		process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via kill (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used when unspecified.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to specify the shell in the securityContext field.

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].volumeDevices`

Description

volumeDevices is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].volumeDevices[]`

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	devicePath is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.ephemeralContainers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.hostAliases`

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

`array`

`.spec.workloads[].template.spec.hostAliases[]`

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type

object

Required

ip

Property	Type	Description
hostnames	array	Hostnames for the above IP address.
ip	string	IP address of the host file entry.

.spec.workloads[].template.spec.hostAliases[].hostnames**Description**

Hostnames for the above IP address.

Type

array

.spec.workloads[].template.spec.hostAliases[].hostnames
[]**Type**

string

.spec.workloads[].template.spec.imagePullSecrets**Description**

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info:

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

Type

array

`.spec.workloads[].template.spec.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.initContainers`

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init

containers cannot currently be added or removed. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

Type

array

```
.spec.workloads[].template.spec.initContainers[]
```

Description

A single application container that you want to run within a pod.

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" to produce a string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" to produce a string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

Property	Type	Description
		the input string will be unchanged. Double \$\$ are single \$, which allows for escaping the \$(VAR_NAME) i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Can be updated. More info: https://kubernetes.io/docs/tasks/configure-pod-container/define-command-argument-container-command-in-a-shell/
env	array	List of environment variables to set in the container. Can be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must match the C_IDENTIFIER. All invalid keys will be reported as an error when the container is starting. When a key exists in multiple sources, the value associated with the last source with the same key will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images/ field is optional to allow higher level config management to default or override container images in workload manifests and StatefulSets.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images/#image-pull-policy

Property	Type	Description
		https://kubernetes.io/docs/concepts/containers/images ↗
<code>lifecycle</code>	<code>object</code>	Actions that the management system should take in response to container lifecycle events. Cannot be updated.
<code>livenessProbe</code>	<code>object</code>	Periodic probe of container liveness. Container v1.4.0+ has a default liveness probe which fails if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
<code>name</code>	<code>string</code>	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
<code>ports</code>	<code>array</code>	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. You must specify a port here to allow the container to be accessible from the network. Most containers have a single port in which they listen on the default "0.0.0.0" address. Multiple ports are possible if the container has multiple listeners. An array with strategic merge patch may corrupt the information. See https://github.com/kubernetes/kubernetes/issues/20380 for more details. Cannot be updated.
<code>readinessProbe</code>	<code>object</code>	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗

Property	Type	Description
		https://kubernetes.io/docs/concepts/workloads/pod-lifecycle#container-probes ↗
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	Compute Resources required by this container. (Can be updated. More info: https://kubernetes.io/docs/concepts/configuration/resource-limits)
<code>restartPolicy</code>	<code>string</code>	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers, this field is not specified, the restart behavior is controlled by the Pod's restart policy and the container type. Setting RestartPolicy to "Always" for the init container will have the following effect: this init container will be continually restarted until all regular containers have terminated. Once all regular containers have completed, all init containers with "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar". Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container after any startupProbe has successfully completed.
<code>securityContext</code>	<code>object</code>	SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/concepts/security/pod-security-context/

Property	Type	Description
		https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ ↗
<code>startupProbe</code>	<code>object</code>	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully. If this probe fails, the Pod will be restarted. This can be used to probe parameters at the beginning of a Pod's life cycle. This cannot be updated. If specified, no other probes are executed until this probe successfully. If this probe fails, the Pod will be restarted. This can be used to probe parameters at the beginning of a Pod's life cycle. This cannot be updated. might take a long time to load data or warm a cache before reaching a steady-state operation. This cannot be updated. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data in its container runtime. If this is not set, reads from stdin within the container will always result in EOF. Default is false.
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When set to true, the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is only open until the first client attaches to stdin, which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process that opens stdin from stdin will never receive an EOF. Default is false.
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container's termination message will be written is mounted in the container's filesystem. Message written is intended to be consumed by a downstream process.

Property	Type	Description
		final status, such as an assertion failure message truncated by the node if greater than 4096 bytes. message length across all containers will be limited. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. Will use the contents of terminationMessagePath if available, otherwise will use the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for stdin. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used within the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the container runtime's default will be used, which might be configured for the container image. Cannot be updated.

`.spec.workloads[].template.spec.initContainers[].args`

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$$(VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.initContainers[].args[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].command`

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$$(VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.initContainers[].command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.initContainers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. Must be a C_IDENTIFIER.
<code>value</code>	<code>string</code>	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input

Property	Type	Description
		string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

object

Required

resource

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>int</code>	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.initContainers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.initContainers[].envFrom`



Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	object	The ConfigMap to select from
<code>prefix</code>	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	object	The Secret to select from

`.spec.workloads[].template.spec.initContainers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
<code>name</code>	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap must be defined

`.spec.workloads[].template.spec.initContainers[].envFrom[].secretRef`

Description

The Secret to select from

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.initContainers[].lifecycle`

Description

Actions that the management system should take in response to container lifecycle events.

Cannot be updated.

Type

object

Property	Type	Description
postStart	object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks
preStop	object	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the

container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.
httpGet	object	HTTPGet specifies the http request to perform.
sleep	object	Sleep represents the duration that the container should sleep before being terminated.
tcpSocket	object	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

**.spec.workloads[].template.spec.initContainers[].lifecycle.
postStart.httpGet.httpHeaders**

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec.command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.

Property	Type	Description
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when

tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.workloads[].template.spec.initContainers[].livenessProbe

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimum
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period in seconds after the processes run

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sigi value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u:
timeoutSeconds	integer	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. <a href="https://kubernetes.io/docs/concepts/worklo
lifecycle#container-probes">https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.workloads[].template.spec.initContainers[].livenessProbe.exec

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`**Required**`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type`object`**Required**`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].ports`

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.initContainers[].readinessProbe`

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used when specified.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].readinessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.initContainers[].resource` **S**

Description

Compute Resources required by this container. Cannot be updated. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.initContainers[].resource` `s.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].resource` `s.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.initContainers[].resource` `s.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.initContainers[].resource` `s.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.initContainers[].securityContext`

Context

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Type

`object`

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.initContainers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.initContainers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.initContainers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.initContainers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.initContainers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-life-cycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used when specified.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].volumeDevice`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.initContainers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain '..'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.nodeSelector`

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info:

<https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

`object`

`.spec.workloads[].template.spec.os`

Description

Specifies the OS of the containers in the pod. Some pod and container fields are restricted if this is set. If the OS field is set to linux, the following fields must be unset: - securityContext.windowsOptions If the OS field is set to windows, following fields must be unset: - spec.hostPID - spec.hostIPC - spec.hostUsers - spec.securityContext.appArmorProfile - spec.securityContext.seLinuxOptions - spec.securityContext.seccompProfile - spec.securityContext.fsGroup - spec.securityContext.fsGroupChangePolicy - spec.securityContext.sysctls - spec.shareProcessNamespace - spec.securityContext.runAsUser - spec.securityContext.runAsGroup - spec.securityContext.supplementalGroups - spec.securityContext.supplementalGroupsPolicy - spec.containers[*].securityContext.appArmorProfile - spec.containers[*].securityContext.seLinuxOptions - spec.containers[*].securityContext.seccompProfile - spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readOnlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsUser - spec.containers[*].securityContext.runAsGroup

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration ↗ Clients should expect to handle additional values and treat unrecognized values in this field as os: null

`.spec.workloads[].template.spec.overhead`

Description

Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info:

<https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md>

Type

object

`.spec.workloads[].template.spec.readinessGates`

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type

array

`.spec.workloads[].template.spec.readinessGates[]`

Description

PodReadinessGate contains the reference to a pod condition

Type

object

Required

conditionType

Property	Type	Description
<code>conditionType</code>	<code>string</code>	ConditionType refers to a condition in the pod's condition list with matching type.

`.spec.workloads[].template.spec.resourceClaims`

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.

Type

`array`

`.spec.workloads[].template.spec.resourceClaims[]`

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod. It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name uniquely identifies this resource claim inside the pod. This must be a

Property	Type	Description
		DNS_LABEL.
resourceClaimName	string	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.
resourceClaimTemplateName	string	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in pod.status.resourceClaimStatuses. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

`.spec.workloads[].template.spec.schedulingGates`

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

array

`.spec.workloads[].template.spec.schedulingGates[]`

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the scheduling gate. Each scheduling gate must have a unique name field.

`.spec.workloads[].template.spec.securityContext`

Description

SecurityContext holds pod-level security attributes and common container settings.
Optional: Defaults to empty. See type description for default values of each field.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>fsGroup</code>	<code>integer</code>	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>fsGroupChangePolicy</code>	<code>string</code>	<code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and

Property	Type	Description
		"Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that

Property	Type	Description
		container. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
supplementalGroups	array	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the

Property	Type	Description
		supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
supplementalGroupsPolicy	string	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
sysctls	array	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	object	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.securityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext.

If set in both `SecurityContext` and `PodSecurityContext`, the value specified in `SecurityContext` takes precedence for that container. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

object

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.securityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.securityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

`array`

`.spec.workloads[].template.spec.securityContext.supplementalGroups[]`

Type

integer

`.spec.workloads[].template.spec.securityContext.sysctls`

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

`.spec.workloads[].template.spec.securityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.workloads[].template.spec.securityContext.windows

Options

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	<code>GMSACredentialSpecName</code> is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	<code>HostProcess</code> determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective <code>HostProcess</code> value (it is not allowed to have a mix of <code>HostProcess</code> containers and non- <code>HostProcess</code> containers). In addition, if <code>HostProcess</code> is true then <code>HostNetwork</code> must also be set to true.

Property	Type	Description
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.tolerations`

Description

If specified, the pod's tolerations.

Type

`array`

`.spec.workloads[].template.spec.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.workloads[].template.spec.topologySpreadConstraints`

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

`array`

`.spec.workloads[].template.spec.topologySpreadConstraints[]`

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

`object`

Required

`maxSkew``topologyKey``whenUnsatisfiable`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values

Property	Type	Description
		from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).
<code>maxSkew</code>	<code>integer</code>	MaxSkew describes the degree to which pods may be unevenly distributed. When <code>whenUnsatisfiable=DoNotSchedule</code>, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1).

Property	Type	Description
		if MaxSkew is 2, incoming pod can be scheduled onto any zone. When <code>whenUnsatisfiable=ScheduleAnyway</code> , it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.

Property	Type	Description
<code>minDomains</code>	<code>integer</code>	MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule. For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: zone1 zone2 zone3 P P P P P P The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.
<code>nodeAffinityPolicy</code>	<code>string</code>	NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are:

Property	Type	Description
		- Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. - Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations. If this value is nil, the behavior is equivalent to the Honor policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
nodeTaintsPolicy	string	NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: - Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. - Ignore: node taints are ignored. All nodes are included. If this value is nil, the behavior is equivalent to the Ignore policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
topologyKey	string	TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket.

Property	Type	Description
		We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of nodeAffinityPolicy and nodeTaintsPolicy. e.g. If TopologyKey is "kubernetes.io/hostname", each Node is a domain of that topology. And, if TopologyKey is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.
whenUnsatisfiable	string	WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. - DoNotSchedule (default) tells the scheduler not to schedule it. - ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: zone1 zone2 zone3 P P P P P If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it <i>more</i> imbalanced. It's a required field.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector`

Description

LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.topologySpreadConstraints[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].matchLabelKeys[]`

Type

`string`

`.spec.workloads[].template.spec.volumes`

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

`array`

`.spec.workloads[].template.spec.volumes[]`

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

`object`

Required

`name`

Property	Type	Description
<code>awsElasticBlockStore</code>	<code>object</code>	<code>awsElasticBlockStore</code> represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. https://kubernetes.io/docs/concepts/storage/volumes/#aws-elastic-block-store

Property	Type	Description
<code>azureDisk</code>	<code>object</code>	azureDisk represents an Azure Data Disk mount on mount to the pod.
<code>azureFile</code>	<code>object</code>	azureFile represents an Azure File Service mount or mount to the pod.
<code>cephfs</code>	<code>object</code>	cephFS represents a Ceph FS mount on the host th lifetime
<code>cinder</code>	<code>object</code>	cinder represents a cinder volume attached and mot machine. More info: https://examples.k8s.io/mysql-c
<code>configMap</code>	<code>object</code>	configMap represents a configMap that should popu
<code>csi</code>	<code>object</code>	csi (Container Storage Interface) represents ephem handled by certain external CSI drivers (Beta feature
<code>downwardAPI</code>	<code>object</code>	downwardAPI represents downward API about the p populate this volume

Property	Type	Description
<code>emptyDir</code>	<code>object</code>	<code>emptyDir</code> represents a temporary directory that shares its lifecycle with the pod. More info: https://kubernetes.io/docs/concepts/storage/emptydir
<code>ephemeral</code>	<code>object</code>	<code>ephemeral</code> represents a volume that is handled by a <code>PersistentVolumeClaim</code>. The volume's lifecycle is tied to the pod that defines it: it is created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod exists, b) normal volumes like restoring from snapshot or capacity are not needed, c) the storage driver is specified through a <code>storageClassName</code> if the storage driver supports dynamic volume provisioning. <code>ephemeral</code> is a <code>PersistentVolumeClaim</code> (see <code>EphemeralVolumeSource</code> for more information on the connection between this volume type and <code>PersistentVolumeClaim</code>). Use <code>PersistentVolumeClaim</code> or one of the vendor-specific <code>StorageClass</code> volumes that persist for longer than the lifecycle of a pod. Use CSI for light-weight local ephemeral volumes if it is not meant to be used that way - see the documentation for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.
<code>fc</code>	<code>object</code>	<code>fc</code> represents a Fibre Channel resource that is attached to a host machine and then exposed to the pod.

Property	Type	Description
<code>flexVolume</code>	<code>object</code>	<code>flexVolume</code> represents a generic volume resource that is provisioned/attached using an exec based plugin.
<code>flocker</code>	<code>object</code>	<code>flocker</code> represents a Flocker volume attached to a Kubernetes node. This depends on the Flocker control service being running on the node.
<code>gcePersistentDisk</code>	<code>object</code>	<code>gcePersistentDisk</code> represents a GCE Disk resource that is provisioned on kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes/#gce-persistent-disk
<code>gitRepo</code>	<code>object</code>	<code>gitRepo</code> represents a git repository at a particular revision. <code>GitRepo</code> is deprecated. To provision a container with a git repo, use an <code>EmptyDir</code> into an <code>InitContainer</code> that clones the repo and then expose the <code>EmptyDir</code> into the Pod's container.
<code>glusterfs</code>	<code>object</code>	<code>glusterfs</code> represents a Glusterfs mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/glusterfs/
<code>hostPath</code>	<code>object</code>	<code>hostPath</code> represents a pre-existing file or directory on the host that is directly exposed to the container. This is generally used for logging agents or other privileged things that are allowed to see the host's file system. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes/#host-path

Property	Type	Description
		image represents an OCI object (a container image mounted on the kubelet's host machine. The volume startup depending on which PullPolicy value is provided. • Always: the kubelet always attempts to pull the reference, even if it already exists locally. Container creation will fail if the pull fails. • Never: the kubelet never pulls the reference and only uses the image or artifact if it is already present on the node. • IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and pull fails. The volume gets re-resolved if the pod gets deleted and then recreated. This means that new remote content will become available on the node. Failure to resolve or pull the image during pod startup may prevent the pod from starting and may add significant latency. Failure to pull the image results in a normal volume backoff and will be reported on the pod status message. The types of objects that may be mounted are determined by the container runtime implementation on the node. The minimum must include all valid types supported by the container runtime. The OCI object gets mounted in a single direct mount (spec.containers[].volumeMounts.mountPath) by the container runtime in the same way as for container images. The volume is mounted read-only (ro) and non-executable files (noexec). mounts for containers are not supported (spec.containers[].volumeMounts.subpath). The field spec.securityContext.fsGroupChangePolicy has no effect on this type.

Property	Type	Description
iscsi	object	iscsi represents an ISCSI Disk resource that is attached to the host machine and then exposed to the pod. More info: https://examples.k8s.io/volumes/iscsi/README.md
name	string	name of the volume. Must be a DNS_LABEL and unique within the namespace. More info: https://kubernetes.io/docs/concepts/overview/objects/names/#names
nfs	object	nfs represents an NFS mount on the host that shares a pod-readable directory. More info: https://kubernetes.io/docs/concepts/storage/volumes/#nfs
persistentVolumeClaim	object	persistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes/#persistentvolumeclaims
photonPersistentDisk	object	photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine
portworxVolume	object	portworxVolume represents a portworx volume attached and mounted on kubelets host machine
projected	object	projected items for all in one resources secrets, configmaps, downward API

Property	Type	Description
<code>quobyte</code>	<code>object</code>	quobyte represents a Quobyte mount on the host throughout the pod's lifetime
<code>rbd</code>	<code>object</code>	rbd represents a Rados Block Device mount on the host throughout the pod's lifetime. More info: https://examples.k8s.io/volumes/rbd/README.md
<code>scaleIO</code>	<code>object</code>	scaleIO represents a ScaleIO persistent volume attached to the pod. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes/scaleio
<code>secret</code>	<code>object</code>	secret represents a secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret
<code>storageos</code>	<code>object</code>	storageOS represents a StorageOS volume attached to the pod. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes/storageos
<code>vsphereVolume</code>	<code>object</code>	vsphereVolume represents a vSphere volume attached to the pod. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes/vsphere-volume

`.spec.workloads[].template.spec.volumes[].awsElasticBlockStore`

Description

awsElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes/elasticblockstore>

https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
partition	integer	partition is the partition in the volume that you want to mount. If not specified, the default is to mount by volume name. Examples: For volume "/dev/sda1", you specify the partition as "1". Similarly, the volume partition for "/dev/sda" is "0" (or you can leave the property empty).
readOnly	boolean	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

`.spec.workloads[].template.spec.volumes[].azureDisk`

Description

azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

object

Required

diskName

diskURI

Property	Type	Description
<code>cachingMode</code>	<code>string</code>	cachingMode is the Host Caching mode: None, Read Only, Read Write.
<code>diskName</code>	<code>string</code>	diskName is the Name of the data disk in the blob storage
<code>diskURI</code>	<code>string</code>	diskURI is the URI of data disk in the blob storage
<code>fsType</code>	<code>string</code>	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>kind</code>	<code>string</code>	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> Defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .

`.spec.workloads[].template.spec.volumes[].azureFile`

Description

`azureFile` represents an Azure File Service mount on the host and bind mount to the pod.

Type

`object`

Required

`secretName`

`shareName`

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .
<code>secretName</code>	<code>string</code>	<code>secretName</code> is the name of secret that contains Azure Storage Account Name and Key
<code>shareName</code>	<code>string</code>	<code>shareName</code> is the azure share Name

`.spec.workloads[].template.spec.volumes[].cephfs`

Description

`cephFS` represents a Ceph FS mount on the host that shares a pod's lifetime

Type

object

Required

monitors

Property	Type	Description
<code>monitors</code>	<code>array</code>	<code>monitors</code> is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>path</code>	<code>string</code>	<code>path</code> is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>secretFile</code>	<code>string</code>	<code>secretFile</code> is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

Property	Type	Description
<code>secretRef</code>	<code>object</code>	secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>user</code>	<code>string</code>	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

`.spec.workloads[].template.spec.volumes[].cephfs.monitors`

Description

monitors is Required: Monitors is a collection of Ceph monitors More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

`array`

`.spec.workloads[].template.spec.volumes[].cephfs.monitors[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].cephfs.secretRef`

Description

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.workloads[].template.spec.volumes[].cinder

Description

cinder represents a cinder volume attached and mounted on kubelets host machine. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

Type

object

Required

volumeID

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code>. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is optional: points to a secret object containing parameters used to connect to OpenStack.
<code>volumeID</code>	<code>string</code>	<code>volumeID</code> used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

`.spec.workloads[].template.spec.volumes[].cinder.secretRef`

Description

`secretRef` is optional: points to a secret object containing parameters used to connect to OpenStack.

Type

`object`

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].configMap`

Description

configMap represents a configMap that should populate this volume

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified

Property	Type	Description
		paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

`.spec.workloads[].template.spec.volumes[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].csi`

Description

csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Type

object

Required

driver

Property	Type	Description
driver	string	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
fsType	string	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
nodePublishSecretRef	object	nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.
readOnly	boolean	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).

Property	Type	Description
<code>volumeAttributes</code>	<code>object</code>	<code>volumeAttributes</code> stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

`.spec.workloads[].template.spec.volumes[].csi.nodePublishSecretRef`

Description

`nodePublishSecretRef` is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI `NodePublishVolume` and `NodeUnpublishVolume` calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.spec.workloads[].template.spec.volumes[].csi.volumeAttributes`

Description

volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

object

.spec.workloads[].template.spec.volumes[].downwardAPI

Description

downwardAPI represents downward API about the pod that should populate this volume

Type

object

Property	Type	Description
defaultMode	integer	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	Items is a list of downward API volume file

**.spec.workloads[].template.spec.volumes[].downwardAPI.
items**

Description

Items is a list of downward API volume file

Type

array

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
<code>fieldRef</code>	<code>object</code>	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
<code>mode</code>	<code>integer</code>	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
<code>path</code>	<code>string</code>	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[].resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>string</code>	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.volumes[].emptyDir`

Description

emptyDir represents a temporary directory that shares a pod's lifetime. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#emptydir>

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
sizeLimit		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.workloads[].template.spec.volumes[].ephemeral

Description

ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see

EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual pod. Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.

Type

object

Property	Type	Description
<code>volumeClaimTemplate</code>	object	Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be <code><pod name>-<volume name></code> where <code><volume name></code> is the name from the <code>PodSpec.Volumes</code> array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will <i>not</i> be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created.

Property	Type	Description
		Required, must not be nil.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate`

Description

Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be `<pod name>-<volume name>` where `<volume name>` is the name from the `PodSpec.Volumes` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

Type

object

Required

spec

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code>	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.

Property	Type	Description
<code>spec</code>	<code>object</code>	The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec`

Description

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

Type

`object`

Property	Type	Description
<code>accessModes</code>	<code>array</code>	accessModes contains the desired access mode volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
<code>dataSource</code>	<code>object</code>	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can support the specified data source, it will create a new

Property	Type	Description
		volume based on the contents of the specific source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If dataSourceRef.namespace is specified, then dataSourceRef contents will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-core API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the name of the specified object matches some installed volume populator or dynamic provisioner. This field will have the functionality of the dataSource field and as such, if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: • While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves them.

Property	Type	Description
		values, and generates an error if a disallowe value is specified. While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
resources	object	resources represents the minimum resources th volume should have. If RecoverVolumeExpansionFailure feature is ena users are allowed to specify resource requirem that are lower than previous value but must still higher than capacity recorded in the status field claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources
selector	object	selector is a label query over volumes to consid binding.
storageClassName	string	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that the <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string value is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClassName</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ [↗] (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]`

Type

string

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef`

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.

(Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:
<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector`

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]

Type

`string`

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.workloads[].template.spec.volumes[].fc`

Description

fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Type

`object`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>lun</code>	<code>integer</code>	lun is Optional: FC target lun number
<code>readOnly</code>	<code>boolean</code>	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in

Property	Type	Description
		VolumeMounts.
<code>targetWWNs</code>	<code>array</code>	targetWWNs is Optional: FC target worldwide names (WWNs)
<code>wwids</code>	<code>array</code>	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

`.spec.workloads[].template.spec.volumes[].fc.targetWWNs`

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

`array`

`.spec.workloads[].template.spec.volumes[].fc.targetWWNs`

`[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].fc.wwids`

Description

wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

Type

`array`

`.spec.workloads[].template.spec.volumes[].fc.wwid[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].flexVolume`

Description

flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type

`object`

Required

`driver`

Property	Type	Description
<code>driver</code>	<code>string</code>	driver is the name of the driver to use for this volume.
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
<code>options</code>	<code>object</code>	options is Optional: this field holds extra command options if any.

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> is Optional: defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code>.
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is Optional: <code>secretRef</code> is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

`.spec.workloads[].template.spec.volumes[].flexVolume.options`

Description

`options` is Optional: this field holds extra command options if any.

Type

`object`

`.spec.workloads[].template.spec.volumes[].flexVolume.secretRef`

Description

`secretRef` is Optional: `secretRef` is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified.

If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].flocker`

Description

flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Type

`object`

Property	Type	Description
<code>datasetName</code>	<code>string</code>	datasetName is Name of the dataset stored as metadata - > name on the dataset for Flocker should be considered as deprecated
<code>datasetUUID</code>	<code>string</code>	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

`.spec.workloads[].template.spec.volumes[].gcePersistentDisk`

Description

`gcePersistentDisk` represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

Type

object

Required

pdName

Property	Type	Description
<code>fsType</code>	string	<code>fsType</code> is filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
<code>partition</code>	integer	<code>partition</code> is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume <code>/dev/sda1</code>, you specify the partition as "1". Similarly, the partition for <code>/dev/sda</code> is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
<code>pdName</code>	string	<code>pdName</code> is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis

`.spec.workloads[].template.spec.volumes[].gitRepo`

Description

gitRepo represents a git repository at a particular revision. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
<code>directory</code>	<code>string</code>	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
<code>repository</code>	<code>string</code>	repository is the URL

Property	Type	Description
<code>revision</code>	<code>string</code>	revision is the commit hash for the specified revision.

`.spec.workloads[].template.spec.volumes[].glusterfs`

Description

glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/glusterfs/README.md>

Type

`object`

Required

`endpoints`

`path`

Property	Type	Description
<code>endpoints</code>	<code>string</code>	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
<code>path</code>	<code>string</code>	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

`.spec.workloads[].template.spec.volumes[].hostPath`

Description

`hostPath` represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

`.spec.workloads[].template.spec.volumes[].image`

Description

image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided:

- Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails.
- Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present.
- IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (`spec.containers[*].volumeMounts.mountPath`) by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (`spec.containers[*].volumeMounts.subpath`). The field `spec.securityContext.fsGroupChangePolicy` has no effect on this volume type.

Type

object

Property	Type	Description
<code>pullPolicy</code>	<code>string</code>	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.
<code>reference</code>	<code>string</code>	Required: Image or artifact reference to be used. Behaves in the same way as <code>pod.spec.containers[*].image</code>. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

`.spec.workloads[].template.spec.volumes[].iscsi`

Description

iscsi represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://examples.k8s.io/volumes/iscsi/README.md>

Type

`object`

Required

`iqn`

`lun`

`targetPortal`

Property	Type	Description
<code>chapAuthDiscovery</code>	<code>boolean</code>	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
<code>chapAuthSession</code>	<code>boolean</code>	chapAuthSession defines whether support iSCSI Session CHAP authentication
<code>fsType</code>	<code>string</code>	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
<code>initiatorName</code>	<code>string</code>	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
<code>iqn</code>	<code>string</code>	iqn is the target iSCSI Qualified Name.
<code>iscsiInterface</code>	<code>string</code>	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
<code>lun</code>	<code>integer</code>	lun represents iSCSI Target Lun number.

Property	Type	Description
<code>portals</code>	<code>array</code>	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
<code>secretRef</code>	<code>object</code>	secretRef is the CHAP Secret for iSCSI target and initiators authentication
<code>targetPortal</code>	<code>string</code>	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

`.spec.workloads[].template.spec.volumes[].iscsi.portals`

Description

portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

Type

`array`

`.spec.workloads[].template.spec.volumes[].iscsi.portals[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].iscsi.secretRef`

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].nfs`

Description

nfs represents an NFS mount on the host that shares a pod's lifetime More info:
<https://kubernetes.io/docs/concepts/storage/volumes#nfs>

Type

object

Required

`path`

`server`

Property	Type	Description
<code>path</code>	<code>string</code>	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗
<code>server</code>	<code>string</code>	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

`.spec.workloads[].template.spec.volumes[].persistentVolumeClaim`

Description

`persistentVolumeClaimVolumeSource` represents a reference to a `PersistentVolumeClaim` in the same namespace. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

`object`

Required

`claimName`

Property	Type	Description
<code>claimName</code>	<code>string</code>	claimName is the name of a <code>PersistentVolumeClaim</code> in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> Will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> . Default <code>false</code> .

`.spec.workloads[].template.spec.volumes[].photonPersistentDisk`

Description

`photonPersistentDisk` represents a `PhotonController` persistent disk attached and mounted on kubelets host machine

Type

`object`

Required

`pdID`

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>pdID</code>	<code>string</code>	<code>pdID</code> is the ID that identifies Photon Controller persistent disk

`.spec.workloads[].template.spec.volumes[].portworxVolume`

Description

portworxVolume represents a portworx volume attached and mounted on kubelets host machine

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

.spec.workloads[].template.spec.volumes[].projected

Description

projected items for all in one resources secrets, configmaps, and downward API

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	<code>defaultMode</code> are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like <code>fsGroup</code> , and the result can be other mode bits set.
<code>sources</code>	<code>array</code>	<code>sources</code> is the list of volume projections. Each entry in this list handles one source.

`.spec.workloads[].template.spec.volumes[].projected.sources`

Description

`sources` is the list of volume projections. Each entry in this list handles one source.

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[]`

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

`object`

Property	Type	Description
<code>clusterTrustBundle</code>	<code>object</code>	ClusterTrustBundle allows a pod to access the <code>.spec.trustBundle</code> field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.
<code>configMap</code>	<code>object</code>	configMap information about the configMap data to project
<code>downwardAPI</code>	<code>object</code>	downwardAPI information about the downwardAPI data to project
<code>secret</code>	<code>object</code>	secret information about the secret data to project

Property	Type	Description
<code>serviceAccountToken</code>	<code>object</code>	serviceAccountToken is information about the serviceAccountToken data to project

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle`

Description

ClusterTrustBundle allows a pod to access the `.spec.trustBundle` field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.`

Type

`object`

Required

`path`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Property	Type	Description
<code>name</code>	<code>string</code>	Select a single ClusterTrustBundle by object name. Mutually-exclusive with <code>signerName</code> and <code>labelSelector</code> .
<code>optional</code>	<code>boolean</code>	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using <code>name</code> , then the named ClusterTrustBundle is allowed not to exist. If using <code>signerName</code> , then the combination of <code>signerName</code> and <code>labelSelector</code> is allowed to match zero ClusterTrustBundles.
<code>path</code>	<code>string</code>	Relative path from the volume root to write the bundle.
<code>signerName</code>	<code>string</code>	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with <code>name</code> . The contents of all selected ClusterTrustBundles will be unified and deduplicated.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector`

Description

Select all ClusterTrustBundles that match this label selector. Only has effect if `signerName` is set. Mutually-exclusive with `name`. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[ ]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.volumes[].projected.sources[].configMap

Description

configMap information about the configMap data to project

Type

object

Property	Type	Description
<code>items</code>	<code>array</code>	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
<code>optional</code>	<code>boolean</code>	optional specify whether the ConfigMap or its keys must be defined

`.spec.workloads[].template.spec.volumes[].projected.sources[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will

error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
<code>path</code>	<code>string</code>	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI`

Description

downwardAPI information about the downwardAPI data to project

Type

`object`

Property	Type	Description
<code>items</code>	<code>array</code>	Items is a list of DownwardAPIVolume file

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items`

Description

Items is a list of DownwardAPIVolume file

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
<code>fieldRef</code>	<code>object</code>	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
<code>mode</code>	<code>integer</code>	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>path</code>	<code>string</code>	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory,

Property	Type	Description
		requests.cpu and requests.memory) are currently supported.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

object

Required

fieldPath

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor		Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

.spec.workloads[].template.spec.volumes[].projected.sources[].secret

Description

secret information about the secret data to project

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified

Property	Type	Description
		paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	optional field specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].projected.sources[].serviceAccountToken`

Description

serviceAccountToken is information about the serviceAccountToken data to project

Type

object

Required

path

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours.Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

`.spec.workloads[].template.spec.volumes[].quobyte`

Description

quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Type

object

Required

registry

volume

Property	Type	Description
group	string	group to map volume access to Default is no group
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user
volume	string	volume is a string that references an already created Quobyte volume by name.

.spec.workloads[].template.spec.volumes[].rbd

Description

rbd represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/rbd/README.md>

Type

object

Required

image

monitors

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Property	Type	Description
monitors	array	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
user	string	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

.spec.workloads[].template.spec.volumes[].rbd.monitors

Description

monitors is a collection of Ceph monitors. More info:

<https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

array

`.spec.workloads[].template.spec.volumes[].rbd.monitors[]`

Type

string

`.spec.workloads[].template.spec.volumes[].rbd.secretRef`

Description

secretRef is name of the authentication secret for RBDUser. If provided overrides keyring.

Default is nil. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].scaleIO`

Description

scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Type

object

Required

gateway

secretRef

system

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
gateway	string	gateway is the host address of the ScaleIO API Gateway.
protectionDomain	string	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.
sslEnabled	boolean	sslEnabled Flag enable/disable SSL communication with Gateway, default false

Property	Type	Description
<code>storageMode</code>	<code>string</code>	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
<code>storagePool</code>	<code>string</code>	storagePool is the ScaleIO Storage Pool associated with the protection domain.
<code>system</code>	<code>string</code>	system is the name of the storage system as configured in ScaleIO.
<code>volumeName</code>	<code>string</code>	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

`.spec.workloads[].template.spec.volumes[].scaleIO.secretRef`

Description

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty.

Property	Type	Description
		Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].secret`

Description

secret represents a secret that should populate this volume. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#secret>

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume

Property	Type	Description
		setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>optional</code>	<code>boolean</code>	optional field specify whether the Secret or its keys must be defined
<code>secretName</code>	<code>string</code>	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

`.spec.workloads[].template.spec.volumes[].secret.items`

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

`array`

`.spec.workloads[].template.spec.volumes[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

`object`

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.workloads[].template.spec.volumes[].storageos

Description

storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.

Type

object

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
<code>volumeNamespace</code>	<code>string</code>	<code>volumeNamespace</code> specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set <code>VolumeName</code> to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

`.spec.workloads[].template.spec.volumes[].storageos.secretRef`

Description

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].vsphereVolume`

Description

vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Type

object

Required

`volumePath`

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>storagePolicyID</code>	<code>string</code>	<code>storagePolicyID</code> is the storage Policy Based Management (SPBM) profile ID associated with the <code>StoragePolicyName</code> .
<code>storagePolicyName</code>	<code>string</code>	<code>storagePolicyName</code> is the storage Policy Based Management (SPBM) profile name.
<code>volumePath</code>	<code>string</code>	<code>volumePath</code> is the path that identifies vSphere volume vmrk

.status

Description

`ComponentStatus` defines the observed state of Connector

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	<code>Annotations</code> is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is

Property	Type	Description
		roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See:
<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
status	string	Status of the condition, one of True, False, Unknown.

Property	Type	Description
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores`
 - `DELETE` : delete collection of ConnectorsCore
 - `GET` : list objects of kind ConnectorsCore
 - `POST` : create a new ConnectorsCore
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores/{name}`
 - `DELETE` : delete the specified ConnectorsCore
 - `GET` : read the specified ConnectorsCore
 - `PATCH` : partially update the specified ConnectorsCore
 - `PUT` : replace the specified ConnectorsCore
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores/{name}/status`
 - `GET` : read status of the specified ConnectorsCore
 - `PATCH` : partially update status of the specified ConnectorsCore
 - `PUT` : replace status of the specified ConnectorsCore

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores`

HTTP method

DELETE

Description

delete collection of ConnectorsCore

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ConnectorsCore

HTTP responses

HTTP code	Response body
200 - OK	ConnectorsCoreList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ConnectorsCore

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsCore</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsCore</code> schema
201 - Created	<code>ConnectorsCore</code> schema
202 - Accepted	<code>ConnectorsCore</code> schema
401 - Unauthorized	Empty

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores/{name}

HTTP method

DELETE

Description

delete the specified ConnectorsCore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ConnectorsCore

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsCore</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ConnectorsCore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsCore</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ConnectorsCore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsCore</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsCore</code> schema
201 - Created	<code>ConnectorsCore</code> schema
401 - Unauthorized	Empty

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores/{name}/status`

HTTP method

GET

Description

read status of the specified ConnectorsCore

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsCore</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ConnectorsCore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsCore</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectorsCore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	ConnectorsCore schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	ConnectorsCore schema
201 - Created	ConnectorsCore schema
401 - Unauthorized	Empty

ConnectorsGit

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsGit is the Schema for the ConnectorsGits API.

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsGitSpec defines the desired state of ConnectorsGit.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsGitSpec defines the desired state of ConnectorsGit.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template	object	Template describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.spec.workloads[].template

Description

Template describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

Type

object

Property	Type	Description
metadata	ObjectMeta ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
spec	object	Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status ↗

.spec.workloads[].template.spec

Description

Specification of the desired behavior of the pod. More info:

<https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

Type

object

Required

containers

Property	Type	Description
activeDeadlineSeconds	integer	Optional duration in seconds the pod may StartTime before the system will actively tr containers. Value must be a positive intege

Property	Type	Description
<code>affinity</code>	<code>object</code>	If specified, the pod's scheduling constrain
<code>automountServiceAccountToken</code>	<code>boolean</code>	AutomountServiceAccountToken indicates be automatically mounted.
<code>containers</code>	<code>array</code>	List of containers belonging to the pod. Co removed. There must be at least one conta
<code>dnsConfig</code>	<code>object</code>	Specifies the DNS parameters of a pod. Pa to the generated DNS configuration based
<code>dnsPolicy</code>	<code>string</code>	Set DNS policy for the pod. Defaults to "Cl" 'ClusterFirstWithHostNet', 'ClusterFirst', 'De in DNSConfig will be merged with the polic DNS options set along with hostNetwork, y to 'ClusterFirstWithHostNet'.
<code>enableServiceLinks</code>	<code>boolean</code>	EnableServiceLinks indicates whether info injected into pod's environment variables, r Optional: Defaults to true.

Property	Type	Description
<code>ephemeralContainers</code>	<code>array</code>	List of ephemeral containers run in this pod. Ephemeral containers are used to perform user-initiated actions on an existing pod to perform user-initiated actions. Ephemeral containers cannot be specified when creating a pod, and must be added to the pod spec. In order to add an ephemeral container to a pod's ephemeralcontainers subresource.
<code>hostAliases</code>	<code>array</code>	HostAliases is an optional list of hosts and their corresponding IP addresses. Hosts must have entries in the /etc/hosts file to be resolved.
<code>hostIPC</code>	<code>boolean</code>	Use the host's ipc namespace. Optional: Default to false
<code>hostNetwork</code>	<code>boolean</code>	Host networking requested for this pod. Use the host's network stack. When option is set, the ports that will be used must be specified. Default to false.
<code>hostPID</code>	<code>boolean</code>	Use the host's pid namespace. Optional: Default to false
<code>hostUsers</code>	<code>boolean</code>	Use the host's user namespace. Optional: Default to false. If true, user namespaces must be enabled on the host kernel with CAP_SYS_MODULE. \

Property	Type	Description
hostname	string	Specifies the hostname of the Pod If not set to a system-defined value.
imagePullSecrets	array	ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images listed in the podSpec. More info: https://kubernetes.io/docs/concepts/configuration/secret#imagepullsecrets-on-a-pod
initContainers	array	List of initialization containers belonging to the pod. Init containers are designed to allow users to easily set up a pod by preparing the container environment. For example, an init container can install a binary, create a file, set environment variables, or even run a pre-upgrade script. Once all the init containers complete successfully, the pod is ready to start. If an init container fails, the pod is considered to have failed and is handled as a failed pod. An init container or normal container may not have Lifecycle actions, Startup probes, or Readiness probes. The resourceRequirements field is optional for init containers. Limits are applied to init containers in a pod. The podSpec may be added or removed. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/initcontainers/
nodeName	string	nodeName indicates in which node this pod is scheduled. If the node name field is set, the kubelet for this node becomes the candidate for scheduling by the scheduler. This field should not be used to express a requirement to have the pod scheduled on a specific node. https://kubernetes.io/docs/concepts/scheduling/kubernetes-scheduling-configuration/#pod-node/#nodename

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	NodeSelector is a selector which must be 1 which must match a node's labels for the p info: https://kubernetes.io/docs/concepts/ci
<code>os</code>	<code>object</code>	Specifies the OS of the containers in the p restricted if this is set. If the OS field is set to linux, the following f securityContext.windowsOptions If the OS field is set to windows, following 1 • <code>spec.hostPID</code> • <code>spec.hostIPC</code> • <code>spec.hostUsers</code> • <code>spec.securityContext.appArmorProfile</code> • <code>spec.securityContext.seLinuxOptions</code> • <code>spec.securityContext.seccompProfile</code> • <code>spec.securityContext.fsGroup</code> • <code>spec.securityContext.fsGroupChangeP</code> • <code>spec.securityContext.sysctls</code> • <code>spec.shareProcessNamespace</code> • <code>spec.securityContext.runAsUser</code> • <code>spec.securityContext.runAsGroup</code> • <code>spec.securityContext.supplementalGroi</code> • <code>spec.securityContext.supplementalGroi</code> • <code>spec.containers[*].securityContext.app/</code> • <code>spec.containers[*].securityContext.seLi</code> • <code>spec.containers[*].securityContext.secc</code>

Property	Type	Description
		- spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readOnlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsUser - spec.containers[*].securityContext.runAsGroup
overhead	object	Overhead represents the resource overhead of a Pod given RuntimeClass. This field will be auto populated by the RuntimeClass admission controller. If the FeatureGate is enabled, overhead must not be set in Pod admission controller will reject Pod creation if already set. If RuntimeClass is configured for the Pod, it will be set to the value defined in the corresponding RuntimeClass. If not set, it will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/482-pod-overhead
preemptionPolicy	string	PreemptionPolicy is the Policy for preempting existing pods. Never, DoNotPreempt, PreemptLowerPriority. Defaults to Never.
priority	integer	The priority value. Various system components use this field to find the priority of the pod. When Priority Admission Controller is enabled, it enforces the MinPriorityClassName setting this field. The admission controller will reject a pod if MinPriorityClassName is set to PriorityClassName. The higher the value, the higher the priority.
priorityClassName	string	If specified, indicates the pod's priority. "system-cluster-critical" and "system-node-critical" are two special keywords which indicate the highest priorities within the cluster.

Property	Type	Description
		former being the highest priority. Any other PriorityClass object with that name. If not set or zero if there is no default.
<code>readinessGates</code>	<code>array</code>	If specified, all readiness gates will be evaluated when all its containers are ready AND all readiness gates have status equal to "True" More info: https://git.k8s.io/enhancements/keps/sig-network/000001-pod-readiness-gates.md
<code>resourceClaims</code>	<code>array</code>	ResourceClaims defines which ResourceClaims the pod needs before the Pod is allowed to start. The resource claims are consumed by containers which consume them by name. This is an alpha field and requires enabling the ResourceClaims feature gate. This field is immutable.
<code>restartPolicy</code>	<code>string</code>	Restart policy for all containers within the pod. Some contexts, only a subset of those values are allowed. More info: https://kubernetes.io/docs/concepts/workloads/pods/#restart-policy
<code>runtimeClassName</code>	<code>string</code>	RuntimeClassName refers to a RuntimeClass object in the kubelet namespace which should be used to run this pod. If no RuntimeClass is named, the pod will not be run. If unspecified, the default runtime handler will be used, which is an implicit class with the name "default". More info: https://kubernetes.io/docs/concepts/workloads/pods/runtime-handlers/

Property	Type	Description
<code>schedulerName</code>	<code>string</code>	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
<code>schedulingGates</code>	<code>array</code>	SchedulingGates is an opaque list of values that must all be satisfied before the pod is scheduled. If schedulingGates is not empty, the pod will not be scheduled until all of the gates in the list are satisfied. If schedulingGates is empty, the pod will be scheduled immediately. SchedulingGates can only be set at pod creation and cannot be updated afterwards.
<code>securityContext</code>	<code>object</code>	SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See the documentation of the fields for each field.
<code>serviceAccount</code>	<code>string</code>	DeprecatedServiceAccount is a deprecated service account to use to run this pod. Deprecated: Use serviceAccountName instead.
<code>serviceAccountName</code>	<code>string</code>	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

Property	Type	Description
<code>setHostnameAsFQDN</code>	<code>boolean</code>	If true the pod's hostname will be configure leaf name (the default). In Linux containers hostname field of the kernel (the nodename containers, this means setting the registry 'HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Fqdn' to FQDN. If a pod does not have FQDN, then the pod will not have a domainname at all.
<code>shareProcessNamespace</code>	<code>boolean</code>	Share a single process namespace between all containers in this pod. If true, containers will be able to view all other containers in the same pod, and the first process will be assigned PID 1. HostPID and ShareProcessNamespace cannot both be true. Optional: Default to false.
<code>subdomain</code>	<code>string</code>	If specified, the fully qualified Pod hostname will be <code><pod-name>.<subdomain>.<pod-namespace>.svc.cluster.local</code> . If the pod has a domainname, it will not have a domainname at all.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no graceful termination is required). If value is nil, the default grace period will be used instead. The default value is dependent on the version of Kubernetes: <table><tr><th>Kubernetes Version</th><th>Default Grace Period</th></tr><tr><td>1.10.0 and earlier</td><td>30 seconds</td></tr><tr><td>1.11.0 and newer</td><td>30 seconds</td></tr></table> The duration in seconds after the processes running in the pod have sent their termination signal and the time when the processes are still alive after the signal (the "cleanup" time).
<code>tolerations</code>	<code>array</code>	If specified, the pod's tolerations.

Property	Type	Description
<code>topologySpreadConstraints</code>	<code>array</code>	TopologySpreadConstraints describes how across topology domains. Scheduler will spread the constraints. All topologySpreadConstra
<code>volumes</code>	<code>array</code>	List of volumes that can be mounted by co info: https://kubernetes.io/docs/concepts/st

`.spec.workloads[].template.spec.affinity`

Description

If specified, the pod's scheduling constraints

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	<code>object</code>	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	<code>object</code>	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

`.spec.workloads[].template.spec.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; then choose the node(s) with the highest sum.

Property	Type	Description
		sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirement specified by this field is not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement specified by this field ceases to be met at some point during pod execution (e.g. due to update), the system may or may not try to eventually evict the pod from its node.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, `requiredDuringScheduling` affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding `matchExpressions`; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
<code>preference</code>	<code>object</code>	A node selector term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

`e.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields`

Description

A list of node selector requirements by node's fields.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key``operator`

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.

Property	Type	Description
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
<code>nodeSelectorTerms</code>	array	Required. A list of node selector terms. The terms are ORed.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelect`

orTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector

orTerms[].matchFields[].values[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding the weight of each element that is matched by the node. Elements that are not matched by the node will not be counted, but they will still contribute to the sum if they have a non-zero weight. This field may be used to give higher priority to certain nodes (e.g. nodes with certain labels) over others.

Property	Type	Description
		adding "weight" to the sum if the node has p which matches the corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement specified by this field is not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement specified by this field ceases to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm intersected, i.e. all terms must be satisfied.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod

Property	Type	Description
		labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code> . null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this `PodAffinityTerm` matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

`yTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key``operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinit`

yTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelect` or

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchL

abelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace`

`eSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expression specified by this field, it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, anti-affinity expression etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has p

Property	Type	Description
		which matches the corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at the current scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm intersected, i.e. all terms must be satisfied.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

Required. A pod affinity term, associated with the corresponding weight.

object

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key``operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

finityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values
[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

finityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this `PodAffinityTerm` matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismat

chLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

paceSelector.matchExpressions[].values[]

Type

string

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paceSelector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces**

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces[]**

Type

string

`.spec.workloads[].template.spec.containers`

Description

List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[]`

Description

A single application container that you want to run within a pod.

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image if this is not provided. Variable references <code>\$(VAR)</code> expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must match the container's environment.

Property	Type	Description
		C_IDENTIFIER. All invalid keys will be reported when the container is starting. When a key exists in multiple sources, the value associated with the last source takes precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images field is optional to allow higher level config management to default or override container images in workload Deployments and StatefulSets.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images
lifecycle	object	Actions that the management system should take in response to container lifecycle events. Cannot be updated.
livenessProbe	object	Periodic probe of container liveness. Container v1.9.0 adds the ability to probe container liveness by using an HTTP GET or POST on the health path. If the probe fails, the container is restarted. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL).

Property	Type	Description
		Cannot be updated.
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address on the container will be accessible from the network. Must be an array with max length 70. Strategic merge patch may corrupt the data if not specified as a string array. See https://github.com/kubernetes/kubernetes/issues/20380 for more info. Cannot be updated.
readinessProbe	object	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
resizePolicy	array	Resources resize policy for the container.
resources	object	Compute Resources required by this container. Cannot be updated. More info: https://kubernetes.io/docs/concepts/configuration/assigning-resources-containers/
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and in that case the only allowed value is "Always". For non-init containers the only allowed value is "Always". If not specified, the restart policy is "Always".

Property	Type	Description
		Pod's restart policy and the container type. Setting RestartPolicy as "Always" for the init container will have the following effect: this init container will be continuously restarted upon exit until all regular containers have terminated. Once all regular containers have completed, all init containers with RestartPolicy "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar". Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next container starts immediately after this init container completes, after any startupProbe has successfully completed.
<code>securityContext</code>	<code>object</code>	SecurityContext defines the security options the container will be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/
<code>startupProbe</code>	<code>object</code>	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully. If this probe fails, the Pod will be restarted. If the livenessProbe fails. This can be used to provide probe parameters at the beginning of a Pod's life cycle. It might take a long time to load data or warm a cache before entering steady-state operation. This cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data at container runtime. If this is not set, reads from stdin will automatically allocate a buffer.

Property	Type	Description
		container will always result in EOF. Default is false
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When false, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process from stdin will never receive an EOF. Default is false
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container termination message will be written is mounted in the container's filesystem. Message written is intended to be of final status, such as an assertion failure message. Message length across all containers will be limited by the node if greater than 4096 bytes. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. If set to File, will use the contents of terminationMessagePath to output the container status message on both success and failure. If set to FallbackToLogsOnError, FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 2 lines, whichever is smaller. Defaults to File. Cannot be updated.

Property	Type	Description
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be u container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystems updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be co container image. Cannot be updated.

`.spec.workloads[].template.spec.containers[].args`

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$(VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

`array`

`.spec.workloads[].template.spec.containers[].args[]`

Type

string

`.spec.workloads[].template.spec.containers[].command`

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$$(VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.containers[].command[]`

Type

string

`.spec.workloads[].template.spec.containers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	Source for the environment variable's value. Cannot be used if value is not empty.

.spec.workloads[].template.spec.containers[].env[].valueFrom

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

object

Property	Type	Description
<code>configMapKeyRef</code>	object	Selects a key of a ConfigMap.
<code>fieldRef</code>	object	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
<code>resourceFieldRef</code>	object	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	object	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.containers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

object

Required

key

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap or its key must be defined

.spec.workloads[].template.spec.containers[].env[].valueFrom.fieldRef

Description

Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Type

object

Required

fieldPath

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.containers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>int</code>	Specifies the output format of the exposed resources, defaults to "1"

Property	Type	Description
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.containers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key of the secret to select from. Must be a valid secret key.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.containers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[].envFrom[]`

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	<code>object</code>	The ConfigMap to select from
<code>prefix</code>	<code>string</code>	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	<code>object</code>	The Secret to select from

`.spec.workloads[].template.spec.containers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap must be defined

.spec.workloads[].template.spec.containers[].envFrom[].secretRef

Description

The Secret to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.containers[].lifecycle`

Description

Actions that the management system should take in response to container lifecycle events.
Cannot be updated.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
<code>preStop</code>	<code>object</code>	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the

Property	Type	Description
		container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗

`.spec.workloads[].template.spec.containers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec.command[]

Type

string

.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name``value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perform.
<code>sleep</code>	<code>object</code>	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	<code>object</code>	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.tcpSocket`

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].livenessProbe`

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies an action involving a GRP
<code>httpGet</code>	object	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	integer	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes
<code>periodSeconds</code>	integer	How often (in seconds) to perform the prot 10 seconds. Minimum value is 1.
<code>successThreshold</code>	integer	Minimum consecutive successes for the pr considered successful after having failed. I

Property	Type	Description
		Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully terminate upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].livenessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.workloads[].template.spec.containers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].livenessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].ports`

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.containers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.containers[].readinessProbe`

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period duration in seconds after the processes running on the pod are sent a termination signal and the time until the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].readinessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.containers[].readinessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.workloads[].template.spec.containers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.containers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.containers[].resources`

Description

Compute Resources required by this container. Cannot be updated. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.containers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.containers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.containers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.containers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.containers[].securityContext`

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.containers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.containers[].securityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.containers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.containers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.containers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.containers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running on the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via <code>TERM</code> (no opportunity to shut down). This is a beta feature and requires enabling <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. Defaults to 30 seconds. <code>spec.terminationGracePeriodSeconds</code> is used if not set.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/probe-lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].startupProbe`

`exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.containers[].startupProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.containers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.containers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.containers[].startupProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.containers[].volumeDevices[]`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.containers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.containers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain '.'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.dnsConfig`

Description

Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy.

Type

`object`

Property	Type	Description
<code>nameservers</code>	<code>array</code>	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
<code>options</code>	<code>array</code>	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.
<code>searches</code>	<code>array</code>	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

`.spec.workloads[].template.spec.dnsConfig.nameservers`

Description

A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.

Type

`array`

`.spec.workloads[].template.spec.dnsConfig.nameservers[]`

Type

`string`

`.spec.workloads[].template.spec.dnsConfig.options`

Description

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type

array

.spec.workloads[].template.spec.dnsConfig.options[]

Description

PodDNSConfigOption defines DNS resolver options of a pod.

Type

object

Property	Type	Description
name	string	Required.
value	string	

.spec.workloads[].template.spec.dnsConfig.searches

Description

A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

Type

array

.spec.workloads[].template.spec.dnsConfig.searches[]

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers`

Description

List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[]`

Description

An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.

Type

`object`

Required

`name`

Property	Type	Description
<code>args</code>	<code>array</code>	Arguments to the entrypoint. The image's CMD is not provided. Variable references \$(VAR_NAME) using the container's environment. If a variable c

Property	Type	Description
		resolved, the reference in the input string will be Double \$\$ are reduced to a single \$, which allow the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" string literal "\$(VAR_NAME)". Escaped reference expanded, regardless of whether the variable ex Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
<code>command</code>	<code>array</code>	Entrypoint array. Not executed within a shell. The ENTRYPOINT is used if this is not provided. Vari \$(VAR_NAME) are expanded using the containe If a variable cannot be resolved, the reference in will be unchanged. Double \$\$ are reduced to a s allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ Escaped references will never be expanded, reg whether the variable exists or not. Cannot be up https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
<code>env</code>	<code>array</code>	List of environment variables to set in the contain updated.

Property	Type	Description
<code>envFrom</code>	<code>array</code>	List of sources to populate environment variables for the container. The keys defined within a source must match the C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source with the same precedence takes precedence. Values defined by an Env with a duplicate key take precedence. Cannot be updated.
<code>image</code>	<code>string</code>	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images
<code>imagePullPolicy</code>	<code>string</code>	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images
<code>lifecycle</code>	<code>object</code>	Lifecycle is not allowed for ephemeral containers
<code>livenessProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>name</code>	<code>string</code>	Name of the ephemeral container specified as a pod's spec. This name must be unique among all containers, and ephemeral containers.

Property	Type	Description
<code>ports</code>	<code>array</code>	Ports are not allowed for ephemeral containers.
<code>readinessProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	Resources are not allowed for ephemeral containers use spare resources already allocated
<code>restartPolicy</code>	<code>string</code>	Restart policy for the container to manage the restart of each container within a pod. This may only be set on non-ephemeral containers. You cannot set this field on ephemeral containers.
<code>securityContext</code>	<code>object</code>	Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the SecurityContext override the equivalent fields of PodSecurityContext.
<code>startupProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data at container runtime. If this is not set, reads from stdin will always result in EOF. Default is false

Property	Type	Description
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When false, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on pod start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the pod is restarted. If this flag is false, a container process from stdin will never receive an EOF. Default is false
<code>targetContainerName</code>	<code>string</code>	If set, the name of the container from PodSpec to which the ephemeral container targets. The ephemeral container will then run in the namespaces (IPC, PID, etc) of this container. If not set, then the ephemeral container uses the namespaces of the Pod spec. The container runtime must implement support for namespace targeting. If the runtime does not support namespace targeting, the result of setting this field is undefined.
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container's termination message will be written is mounted in the container's filesystem. Message written is intended to be consumed by kubelet. Message length across all containers will be limited by the node if greater than 4096 bytes. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. If set to "FallbackToLogsOnError", the contents of the container logs will be used if the message is not available.

Property	Type	Description
		container status message on both success and failure. FallbackToLogsOnError will use the last chunk of output if the termination message file is empty and exited with an error. The log output is limited to 2048 lines, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for stdin. Requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used within the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be configurable by the container image. Cannot be updated.

`.spec.workloads[].template.spec.ephemeralContainers[].args`

Description

Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `$(foo $(foo $(bar)))`.

"\$\$ (VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].args[]

Type

string

.spec.workloads[].template.spec.ephemeralContainers[].command

Description

Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].command[]

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].env[].`

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. Must be a C_IDENTIFIER.
<code>value</code>	<code>string</code>	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME)

Property	Type	Description
		syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <pre>metadata.labels['<KEY>'], metadata.annotations['<KEY>'], spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.</pre>

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

object

Required

resource

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>		Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[]`

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	The ConfigMap to select from
prefix	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
secretRef	object	The Secret to select from

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[].secretRef`

Description

The Secret to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle`

Description

Lifecycle is not allowed for ephemeral containers.

Type

object

Property	Type	Description
<code>postStart</code>	object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks
<code>preStop</code>	object	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].ifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].ifecycle.postStart.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be

Property	Type	Description
		an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility.

Property	Type	Description
		There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec.command[]

Type

string

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.

Property	Type	Description
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when

tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe`

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u:
timeoutSeconds	integer	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. <a href="https://kubernetes.io/docs/concepts/worklo
lifecycle#container-probes">https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.workloads[].template.spec.ephemeralContainers[].li venessProbe.exec

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`

Required

`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.ephemeralContainers[].li venessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].ports`

Description

Ports are not allowed for ephemeral containers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe`

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.ephemeralContainers[].resources`

Description

Resources are not allowed for ephemeral containers. Ephemeral containers use spare resources already allocated to the pod.

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.ephemeralContainers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.ephemeralContainers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.ephemeralContainers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext`

Description

Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe`

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.
failureThreshold	integer	Minimum consecutive failures for the probe considered failed after having succeeded.

Property	Type	Description
		Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the prot 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tin

Property	Type	Description
		process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via kill (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used when specified.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to specify the shell explicitly with the last argument in the command list. One or more args following the command will be passed through to the command.

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name``value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].volumeDevices[]`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath`

`name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.ephemeralContainers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.hostAliases`

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

`array`

`.spec.workloads[].template.spec.hostAliases[]`

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type`object`**Required**`ip`

Property	Type	Description
<code>hostnames</code>	<code>array</code>	Hostnames for the above IP address.
<code>ip</code>	<code>string</code>	IP address of the host file entry.

`.spec.workloads[].template.spec.hostAliases[].hostnames`**Description**

Hostnames for the above IP address.

Type`array`**`.spec.workloads[].template.spec.hostAliases[].hostnames`
`[]`****Type**`string`**`.spec.workloads[].template.spec.imagePullSecrets`****Description**

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info:

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

Type

array

`.spec.workloads[].template.spec.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.spec.workloads[].template.spec.initContainers`

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init

Property	Type	Description
args	array	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

Property	Type	Description
		the input string will be unchanged. Double \$\$ are single \$, which allows for escaping the \$(VAR_NAME) i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Can be updated. More info: https://kubernetes.io/docs/tasks/configure-pod-container/define-command-argument-container-command-in-a-shell/
env	array	List of environment variables to set in the container. Can be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must match those of the container. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source with the same key will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images/ field is optional to allow higher level config management to default or override container images in workload manifests and StatefulSets.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images/#image-pull-policy

Property	Type	Description
		https://kubernetes.io/docs/concepts/containers/images/ ↗
<code>lifecycle</code>	<code>object</code>	Actions that the management system should take in response to container lifecycle events. Cannot be updated.
<code>livenessProbe</code>	<code>object</code>	Periodic probe of container liveness. Container v1 will be restarted if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
<code>name</code>	<code>string</code>	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
<code>ports</code>	<code>array</code>	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. You must specify a port here to allow the container to be accessible from the network. Most containers are exposed by default to the host as "0.0.0.0". Mounting an array with strategic merge patch may corrupt the data. See https://github.com/kubernetes/kubernetes/issues/108132 for more info. Cannot be updated.
<code>readinessProbe</code>	<code>object</code>	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗

Property	Type	Description
		https://kubernetes.io/docs/concepts/workloads/pod-lifecycle#container-probes ↗
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	Compute Resources required by this container. (Can be updated.) More info: https://kubernetes.io/docs/concepts/configuration/resource-limits
<code>restartPolicy</code>	<code>string</code>	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers, this field is not specified, the restart behavior is controlled by the Pod's restart policy and the container type. Setting RestartPolicy to "Always" for the init container will have the following effect: this init container will be continually restarted until all regular containers have terminated. Once all regular containers have completed, all init containers with "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar". Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container after any startupProbe has successfully completed.
<code>securityContext</code>	<code>object</code>	SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/concepts/security/pod-security-context/

Property	Type	Description
		https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ ↗
<code>startupProbe</code>	<code>object</code>	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully. If this probe fails, the Pod will be restarted. This can be used to probe parameters at the beginning of a Pod's life cycle. This cannot be updated. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data in its container runtime. If this is not set, reads from stdin will always result in EOF. Default is false.
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When set to true, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on pod start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process that reads from stdin will never receive an EOF. Default is false.
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container's termination message will be written is mounted in the container's filesystem. Message written is intended to be consumed by a user-defined process or another

Property	Type	Description
		final status, such as an assertion failure message truncated by the node if greater than 4096 bytes. message length across all containers will be limited. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. Will use the contents of terminationMessagePath if available, otherwise will use the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for stdin. Requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used within the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the container runtime's default will be used, which might be configured for the container image. Cannot be updated.

`.spec.workloads[].template.spec.initContainers[].args`

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$ (VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.initContainers[].args[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].command`

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$ (VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.initContainers[].command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.initContainers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. Must be a C_IDENTIFIER.
<code>value</code>	<code>string</code>	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input

Property	Type	Description
		string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

object

Required

resource

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>int</code>	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.initContainers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.initContainers[].envFrom`



Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	object	The ConfigMap to select from
<code>prefix</code>	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	object	The Secret to select from

`.spec.workloads[].template.spec.initContainers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
<code>name</code>	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap must be defined

`.spec.workloads[].template.spec.initContainers[].envFrom[].secretRef`

Description

The Secret to select from

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.initContainers[].lifecycle`

Description

Actions that the management system should take in response to container lifecycle events.

Cannot be updated.

Type

object

Property	Type	Description
postStart	object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks
preStop	object	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the

container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.
httpGet	object	HTTPGet specifies the http request to perform.
sleep	object	Sleep represents the duration that the container should sleep before being terminated.
tcpSocket	object	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

**.spec.workloads[].template.spec.initContainers[].lifecycle.
postStart.httpGet.httpHeaders**

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec.command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.

Property	Type	Description
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when

tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.workloads[].template.spec.initContainers[].livenessProbe

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sigi value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u:
timeoutSeconds	integer	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.workloads[].template.spec.initContainers[].livenessProbe.exec

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`**Required**`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type`object`**Required**`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].ports`

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.initContainers[].readinessProbe`

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].readinessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.initContainers[].resource` **S**

Description

Compute Resources required by this container. Cannot be updated. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.initContainers[].resource` `s.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].resource` `s.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.initContainers[].resource` `s.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.initContainers[].resource` `s.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.initContainers[].securityContext`

Context

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
<code>procMount</code>	<code>string</code>	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.initContainers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.workloads[].template.spec.initContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.initContainers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.initContainers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.initContainers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.initContainers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].volumeDevices[]`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath`

`name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.initContainers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.nodeSelector`

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info:

<https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

`object`

`.spec.workloads[].template.spec.os`

Description

Specifies the OS of the containers in the pod. Some pod and container fields are restricted if this is set. If the OS field is set to linux, the following fields must be unset: - securityContext.windowsOptions If the OS field is set to windows, following fields must be unset: - spec.hostPID - spec.hostIPC - spec.hostUsers - spec.securityContext.appArmorProfile - spec.securityContext.seLinuxOptions - spec.securityContext.seccompProfile - spec.securityContext.fsGroup - spec.securityContext.fsGroupChangePolicy - spec.securityContext.sysctls - spec.shareProcessNamespace - spec.securityContext.runAsUser - spec.securityContext.runAsGroup - spec.securityContext.supplementalGroups - spec.securityContext.supplementalGroupsPolicy - spec.containers[*].securityContext.appArmorProfile - spec.containers[*].securityContext.seLinuxOptions - spec.containers[*].securityContext.seccompProfile - spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readOnlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsUser - spec.containers[*].securityContext.runAsGroup

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration ↗ Clients should expect to handle additional values and treat unrecognized values in this field as os: null

`.spec.workloads[].template.spec.overhead`

Description

Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info:

<https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md>

Type

object

`.spec.workloads[].template.spec.readinessGates`

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type

array

`.spec.workloads[].template.spec.readinessGates[]`

Description

PodReadinessGate contains the reference to a pod condition

Type

object

Required

conditionType

Property	Type	Description
<code>conditionType</code>	<code>string</code>	ConditionType refers to a condition in the pod's condition list with matching type.

`.spec.workloads[].template.spec.resourceClaims`

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.

Type

`array`

`.spec.workloads[].template.spec.resourceClaims[]`

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod. It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name uniquely identifies this resource claim inside the pod. This must be a

Property	Type	Description
		DNS_LABEL.
<code>resourceClaimName</code>	<code>string</code>	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.
<code>resourceClaimTemplateName</code>	<code>string</code>	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in pod.status.resourceClaimStatuses. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

`.spec.workloads[].template.spec.schedulingGates`

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

array

`.spec.workloads[].template.spec.schedulingGates[]`

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the scheduling gate. Each scheduling gate must have a unique name field.

`.spec.workloads[].template.spec.securityContext`

Description

SecurityContext holds pod-level security attributes and common container settings.
Optional: Defaults to empty. See type description for default values of each field.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>fsGroup</code>	<code>integer</code>	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>fsGroupChangePolicy</code>	<code>string</code>	<code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and

Property	Type	Description
		"Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that

Property	Type	Description
		container. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the

Property	Type	Description
		supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.securityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext.

If set in both `SecurityContext` and `PodSecurityContext`, the value specified in `SecurityContext` takes precedence for that container. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.securityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.securityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

`array`

`.spec.workloads[].template.spec.securityContext.supplementalGroups[]`

Type

integer

`.spec.workloads[].template.spec.securityContext.sysctls`

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

`.spec.workloads[].template.spec.securityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.workloads[].template.spec.securityContext.windows

Options

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	<code>GMSACredentialSpecName</code> is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	<code>HostProcess</code> determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective <code>HostProcess</code> value (it is not allowed to have a mix of <code>HostProcess</code> containers and non- <code>HostProcess</code> containers). In addition, if <code>HostProcess</code> is true then <code>HostNetwork</code> must also be set to true.

Property	Type	Description
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.tolerations`

Description

If specified, the pod's tolerations.

Type

`array`

`.spec.workloads[].template.spec.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.workloads[].template.spec.topologySpreadConstraints`

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[]`

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

object

Required

maxSkew

topologyKey

whenUnsatisfiable

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values

Property	Type	Description
		from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).
maxSkew	integer	MaxSkew describes the degree to which pods may be unevenly distributed. When <code>whenUnsatisfiable=DoNotSchedule</code>, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1).

Property	Type	Description
		if MaxSkew is 2, incoming pod can be scheduled onto any zone. When <code>whenUnsatisfiable=ScheduleAnyway</code> , it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.

Property	Type	Description
<code>minDomains</code>	<code>integer</code>	MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule. For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: zone1 zone2 zone3 P P P P P P The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.
<code>nodeAffinityPolicy</code>	<code>string</code>	NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are:

Property	Type	Description
		- Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. - Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations. If this value is nil, the behavior is equivalent to the Honor policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
nodeTaintsPolicy	string	NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: - Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. - Ignore: node taints are ignored. All nodes are included. If this value is nil, the behavior is equivalent to the Ignore policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
topologyKey	string	TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket.

Property	Type	Description
		We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of nodeAffinityPolicy and nodeTaintsPolicy. e.g. If TopologyKey is "kubernetes.io/hostname", each Node is a domain of that topology. And, if TopologyKey is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.
<code>whenUnsatisfiable</code>	<code>string</code>	WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. • DoNotSchedule (default) tells the scheduler not to schedule it. • ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: zone1 zone2 zone3 P P P P P If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it <i>more</i> imbalanced. It's a required field.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector`

Description

LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.topologySpreadConstraints[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].matchLabelKeys[]`

Type

`string`

`.spec.workloads[].template.spec.volumes`

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

`array`

`.spec.workloads[].template.spec.volumes[]`

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

`object`

Required

`name`

Property	Type	Description
<code>awsElasticBlockStore</code>	<code>object</code>	<code>awsElasticBlockStore</code> represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. https://kubernetes.io/docs/concepts/storage/volumes/#aws-elastic-block-store

Property	Type	Description
<code>azureDisk</code>	<code>object</code>	azureDisk represents an Azure Data Disk mount on mount to the pod.
<code>azureFile</code>	<code>object</code>	azureFile represents an Azure File Service mount or mount to the pod.
<code>cephfs</code>	<code>object</code>	cephFS represents a Ceph FS mount on the host th lifetime
<code>cinder</code>	<code>object</code>	cinder represents a cinder volume attached and mot machine. More info: https://examples.k8s.io/mysql-c
<code>configMap</code>	<code>object</code>	configMap represents a configMap that should popu
<code>csi</code>	<code>object</code>	csi (Container Storage Interface) represents ephem handled by certain external CSI drivers (Beta feature
<code>downwardAPI</code>	<code>object</code>	downwardAPI represents downward API about the p populate this volume

Property	Type	Description
<code>emptyDir</code>	<code>object</code>	<code>emptyDir</code> represents a temporary directory that shares its life cycle with the Pod (the volume is present as long as the Pod is running and is deleted when the Pod is terminated: the data stored in the volume is also deleted). More info: https://kubernetes.io/docs/concepts/storage/emptydir
<code>ephemeral</code>	<code>object</code>	<code>ephemeral</code> represents a volume that is handled by a <code>PersistentVolumeClaim</code> and is tied to the pod that defines it. The volume's lifecycle is tied to the pod that defines it: the volume is created before the pod starts, and deleted when the pod is terminated. This is useful for data that is only needed for the life of the pod, and is not needed after the pod terminates. Use this if: a) the volume is only needed while the pod is running, b) the volume is not needed after the pod terminates, c) the storage driver is specified through a <code>PersistentVolumeClaim</code> (see <code>EphemeralVolumeSource</code> for more information on the connection between this volume type and <code>PersistentVolumeClaim</code>). Use <code>PersistentVolumeClaim</code> or one of the vendor-specific <code>StorageClass</code> volumes that persist for longer than the lifecycle of a pod. Use CSI for light-weight local ephemeral volumes if it is meant to be used that way - see the documentation for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.
<code>fc</code>	<code>object</code>	<code>fc</code> represents a Fibre Channel resource that is attached to a storage machine and then exposed to the pod.

Property	Type	Description
<code>flexVolume</code>	<code>object</code>	<code>flexVolume</code> represents a generic volume resource that is provisioned/attached using an exec based plugin.
<code>flocker</code>	<code>object</code>	<code>flocker</code> represents a Flocker volume attached to a kubelet. This depends on the Flocker control service being running.
<code>gcePersistentDisk</code>	<code>object</code>	<code>gcePersistentDisk</code> represents a GCE Disk resource that is provisioned on kubelet's host machine and then exposed to the pod. https://kubernetes.io/docs/concepts/storage/volumes/#gce-persistent-disk
<code>gitRepo</code>	<code>object</code>	<code>gitRepo</code> represents a git repository at a particular revision. <code>GitRepo</code> is deprecated. To provision a container with a git repo, use an <code>EmptyDir</code> into an <code>InitContainer</code> that clones the repo and then expose the <code>EmptyDir</code> into the Pod's container.
<code>glusterfs</code>	<code>object</code>	<code>glusterfs</code> represents a Glusterfs mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/glusterfs/
<code>hostPath</code>	<code>object</code>	<code>hostPath</code> represents a pre-existing file or directory on the host that is directly exposed to the container. This is generally used for logging agents or other privileged things that are allowed to see the host's file system. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes/#host-path

Property	Type	Description
		image represents an OCI object (a container image mounted on the kubelet's host machine. The volume startup depending on which PullPolicy value is provided. • Always: the kubelet always attempts to pull the reference, even if it already exists locally. Container creation will fail if the pull fails. • Never: the kubelet never pulls the reference and only uses the image or artifact if it is already present on the node. • IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and pull fails. The volume gets re-resolved if the pod gets deleted and then is recreated. This means that new remote content will become available. Failure to resolve or pull the image during pod startup may prevent the pod from starting and may add significant latency. Failure to pull the image results in a normal volume backoff and will be reported on the pod status message. The types of objects that may be mounted are defined by the container runtime implementation on the node. The minimum must include all valid types supported by the container runtime. The OCI object gets mounted in a single direct mount (spec.containers[].volumeMounts.mountPath) by the kubelet. The mount is read-only (ro) and non-executable files (noexec). Mounts for containers are not supported for subpaths (spec.containers[].volumeMounts.subpath). The field spec.securityContext.fsGroupChangePolicy has no effect on this type.

Property	Type	Description
<code>iscsi</code>	<code>object</code>	iscsi represents an ISCSI Disk resource that is attached to the host machine and then exposed to the pod. More info: https://examples.k8s.io/volumes/iscsi/README.md
<code>name</code>	<code>string</code>	name of the volume. Must be a DNS_LABEL and unique within the namespace. More info: https://kubernetes.io/docs/concepts/overview/objects/names/#names
<code>nfs</code>	<code>object</code>	nfs represents an NFS mount on the host that shares a pod-readable directory. More info: https://kubernetes.io/docs/concepts/storage/volumes/#nfs
<code>persistentVolumeClaim</code>	<code>object</code>	persistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes/#persistentvolumeclaims
<code>photonPersistentDisk</code>	<code>object</code>	photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine
<code>portworxVolume</code>	<code>object</code>	portworxVolume represents a portworx volume attached and mounted on kubelets host machine
<code>projected</code>	<code>object</code>	projected items for all in one resources secrets, configmaps, downward API

Property	Type	Description
<code>quobyte</code>	<code>object</code>	quobyte represents a Quobyte mount on the host throughout the pod's lifetime
<code>rbd</code>	<code>object</code>	rbd represents a Rados Block Device mount on the host throughout the pod's lifetime. More info: https://examples.k8s.io/volumes/rbd/README.md
<code>scaleIO</code>	<code>object</code>	scaleIO represents a ScaleIO persistent volume attached to the pod. More info: https://examples.k8s.io/volumes/scaleio/README.md
<code>secret</code>	<code>object</code>	secret represents a secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret
<code>storageos</code>	<code>object</code>	storageOS represents a StorageOS volume attached to the pod. More info: https://examples.k8s.io/volumes/storageos/README.md
<code>vsphereVolume</code>	<code>object</code>	vsphereVolume represents a vSphere volume attached to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#vspherevolume

`.spec.workloads[].template.spec.volumes[].awsElasticBlockStore`

Description

awsElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

<https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
partition	integer	partition is the partition in the volume that you want to mount. If not specified, the default is to mount by volume name. Examples: For volume "/dev/sda1", you specify the partition as "1". Similarly, the volume partition for "/dev/sda" is "0" (or you can leave the property empty).
readOnly	boolean	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

`.spec.workloads[].template.spec.volumes[].azureDisk`

Description

azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

object

Required

diskName

diskURI

Property	Type	Description
<code>cachingMode</code>	<code>string</code>	cachingMode is the Host Caching mode: None, Read Only, Read Write.
<code>diskName</code>	<code>string</code>	diskName is the Name of the data disk in the blob storage
<code>diskURI</code>	<code>string</code>	diskURI is the URI of data disk in the blob storage
<code>fsType</code>	<code>string</code>	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>kind</code>	<code>string</code>	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> Defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .

`.spec.workloads[].template.spec.volumes[].azureFile`

Description

`azureFile` represents an Azure File Service mount on the host and bind mount to the pod.

Type

`object`

Required

`secretName`

`shareName`

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .
<code>secretName</code>	<code>string</code>	<code>secretName</code> is the name of secret that contains Azure Storage Account Name and Key
<code>shareName</code>	<code>string</code>	<code>shareName</code> is the azure share Name

`.spec.workloads[].template.spec.volumes[].cephfs`

Description

`cephFS` represents a Ceph FS mount on the host that shares a pod's lifetime

Type

object

Required

monitors

Property	Type	Description
<code>monitors</code>	<code>array</code>	<code>monitors</code> is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
<code>path</code>	<code>string</code>	<code>path</code> is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
<code>secretFile</code>	<code>string</code>	<code>secretFile</code> is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

Property	Type	Description
<code>secretRef</code>	<code>object</code>	secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>user</code>	<code>string</code>	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

`.spec.workloads[].template.spec.volumes[].cephfs.monitors`

Description

monitors is Required: Monitors is a collection of Ceph monitors More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

`array`

`.spec.workloads[].template.spec.volumes[].cephfs.monitors[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].cephfs.secretRef`

Description

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.workloads[].template.spec.volumes[].cinder

Description

cinder represents a cinder volume attached and mounted on kubelets host machine. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

Type

object

Required

volumeID

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code>. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is optional: points to a secret object containing parameters used to connect to OpenStack.
<code>volumeID</code>	<code>string</code>	<code>volumeID</code> used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

`.spec.workloads[].template.spec.volumes[].cinder.secretRef`

Description

`secretRef` is optional: points to a secret object containing parameters used to connect to OpenStack.

Type

`object`

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].configMap`

Description

configMap represents a configMap that should populate this volume

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified

Property	Type	Description
		paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

`.spec.workloads[].template.spec.volumes[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].csi`

Description

csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Type

object

Required

driver

Property	Type	Description
<code>driver</code>	<code>string</code>	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
<code>fsType</code>	<code>string</code>	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
<code>nodePublishSecretRef</code>	<code>object</code>	nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.
<code>readOnly</code>	<code>boolean</code>	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).

Property	Type	Description
<code>volumeAttributes</code>	<code>object</code>	<code>volumeAttributes</code> stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

`.spec.workloads[].template.spec.volumes[].csi.nodePublishSecretRef`

Description

`nodePublishSecretRef` is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI `NodePublishVolume` and `NodeUnpublishVolume` calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.spec.workloads[].template.spec.volumes[].csi.volumeAttributes`

Description

volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

object

.spec.workloads[].template.spec.volumes[].downwardAPI

Description

downwardAPI represents downward API about the pod that should populate this volume

Type

object

Property	Type	Description
defaultMode	integer	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	Items is a list of downward API volume file

.spec.workloads[].template.spec.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

array

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
<code>fieldRef</code>	<code>object</code>	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
<code>mode</code>	<code>integer</code>	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
<code>path</code>	<code>string</code>	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[].resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>string</code>	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.volumes[].emptyDir`

Description

emptyDir represents a temporary directory that shares a pod's lifetime. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#emptydir>

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
sizeLimit		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.workloads[].template.spec.volumes[].ephemeral

Description

ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see

EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual pod. Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.

Type

object

Property	Type	Description
volumeClaimTemplate	object	Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be <code><pod name>-<volume name></code> where <code><volume name></code> is the name from the <code>PodSpec.Volumes</code> array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will <i>not</i> be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created.

Property	Type	Description
		Required, must not be nil.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate`

Description

Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be ``<pod name>-<volume name>`` where ``<volume name>`` is the name from the ``PodSpec.Volumes`` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

Type

object

Required

spec

Property	Type	Description
metadata	ObjectMeta	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.

Property	Type	Description
<code>spec</code>	<code>object</code>	The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec`

Description

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

Type

`object`

Property	Type	Description
<code>accessModes</code>	<code>array</code>	accessModes contains the desired access mode volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
<code>dataSource</code>	<code>object</code>	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can support the specified data source, it will create a new

Property	Type	Description
		volume based on the contents of the specific source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If dataSourceRef.namespace is specified, then dataSourceRef contents will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-core API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the name of the specified object matches some installed volume populator or dynamic provisioner. This field will take precedence over the dataSource field and as long as both fields are non-empty, they must have the same namespace value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields must have the same namespace. (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same namespace value and must be empty. There are three important differences between dataSource and dataSourceRef: • While dataSource only allows two specific types of objects, dataSourceRef allows any non-core objects as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves them.

Property	Type	Description
		values, and generates an error if a disallowe value is specified. While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
resources	object	resources represents the minimum resources th volume should have. If RecoverVolumeExpansionFailure feature is ena users are allowed to specify resource requirem that are lower than previous value but must still higher than capacity recorded in the status field claim. More info: https://kubernetes.io/docs/concepts/storage/per-volumes#resources
selector	object	selector is a label query over volumes to consid binding.
storageClassName	string	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per-volumes#class-1

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that the <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string value is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClassName</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ ↗ (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]`

Type

string

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef`

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.

(Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:
<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector`

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.workloads[].template.spec.volumes[].fc`

Description

fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Type

`object`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>lun</code>	<code>integer</code>	lun is Optional: FC target lun number
<code>readOnly</code>	<code>boolean</code>	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in

Property	Type	Description
		VolumeMounts.
<code>targetWWNs</code>	<code>array</code>	targetWWNs is Optional: FC target worldwide names (WWNs)
<code>wwids</code>	<code>array</code>	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

`.spec.workloads[].template.spec.volumes[].fc.targetWWNs`

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

`array`

`.spec.workloads[].template.spec.volumes[].fc.targetWWNs`

`[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].fc.wwids`

Description

wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

Type

`array`

`.spec.workloads[].template.spec.volumes[].fc.wwid[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].flexVolume`

Description

flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type

`object`

Required

`driver`

Property	Type	Description
<code>driver</code>	<code>string</code>	driver is the name of the driver to use for this volume.
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
<code>options</code>	<code>object</code>	options is Optional: this field holds extra command options if any.

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> is Optional: defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code>.
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is Optional: <code>secretRef</code> is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

`.spec.workloads[].template.spec.volumes[].flexVolume.options`

Description

`options` is Optional: this field holds extra command options if any.

Type

`object`

`.spec.workloads[].template.spec.volumes[].flexVolume.secretRef`

Description

`secretRef` is Optional: `secretRef` is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified.

If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].flocker`

Description

flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Type

`object`

Property	Type	Description
<code>datasetName</code>	<code>string</code>	datasetName is Name of the dataset stored as metadata - > name on the dataset for Flocker should be considered as deprecated
<code>datasetUUID</code>	<code>string</code>	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

`.spec.workloads[].template.spec.volumes[].gcePersistentDisk`

Description

`gcePersistentDisk` represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

Type

object

Required

pdName

Property	Type	Description
<code>fsType</code>	string	<code>fsType</code> is filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
<code>partition</code>	integer	<code>partition</code> is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
<code>pdName</code>	string	<code>pdName</code> is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis 

`.spec.workloads[].template.spec.volumes[].gitRepo`

Description

gitRepo represents a git repository at a particular revision. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
<code>directory</code>	<code>string</code>	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
<code>repository</code>	<code>string</code>	repository is the URL

Property	Type	Description
<code>revision</code>	<code>string</code>	revision is the commit hash for the specified revision.

`.spec.workloads[].template.spec.volumes[].glusterfs`

Description

glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/glusterfs/README.md>

Type

`object`

Required

`endpoints`

`path`

Property	Type	Description
<code>endpoints</code>	<code>string</code>	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
<code>path</code>	<code>string</code>	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

`.spec.workloads[].template.spec.volumes[].hostPath`

Description

`hostPath` represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

`.spec.workloads[].template.spec.volumes[].image`

Description

image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided: - Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. - Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. - IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (`spec.containers[*].volumeMounts.mountPath`) by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (`spec.containers[*].volumeMounts.subpath`). The field `spec.securityContext.fsGroupChangePolicy` has no effect on this volume type.

Type

object

Property	Type	Description
<code>pullPolicy</code>	<code>string</code>	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.
<code>reference</code>	<code>string</code>	Required: Image or artifact reference to be used. Behaves in the same way as <code>pod.spec.containers[*].image</code>. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

`.spec.workloads[].template.spec.volumes[].iscsi`

Description

iscsi represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://examples.k8s.io/volumes/iscsi/README.md>

Type

`object`

Required

`iqn`

`lun`

`targetPortal`

Property	Type	Description
<code>chapAuthDiscovery</code>	<code>boolean</code>	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
<code>chapAuthSession</code>	<code>boolean</code>	chapAuthSession defines whether support iSCSI Session CHAP authentication
<code>fsType</code>	<code>string</code>	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
<code>initiatorName</code>	<code>string</code>	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
<code>iqn</code>	<code>string</code>	iqn is the target iSCSI Qualified Name.
<code>iscsiInterface</code>	<code>string</code>	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
<code>lun</code>	<code>integer</code>	lun represents iSCSI Target Lun number.

Property	Type	Description
<code>portals</code>	<code>array</code>	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
<code>secretRef</code>	<code>object</code>	secretRef is the CHAP Secret for iSCSI target and initiators authentication
<code>targetPortal</code>	<code>string</code>	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

`.spec.workloads[].template.spec.volumes[].iscsi.portals`

Description

portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

Type

`array`

`.spec.workloads[].template.spec.volumes[].iscsi.portals[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].iscsi.secretRef`

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].nfs`

Description

nfs represents an NFS mount on the host that shares a pod's lifetime More info:
<https://kubernetes.io/docs/concepts/storage/volumes#nfs>

Type

object

Required

`path`

`server`

Property	Type	Description
<code>path</code>	<code>string</code>	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
<code>server</code>	<code>string</code>	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

`.spec.workloads[].template.spec.volumes[].persistentVolumeClaim`

Description

`persistentVolumeClaimVolumeSource` represents a reference to a `PersistentVolumeClaim` in the same namespace. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

`object`

Required

`claimName`

Property	Type	Description
<code>claimName</code>	<code>string</code>	claimName is the name of a <code>PersistentVolumeClaim</code> in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> Will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> . Default <code>false</code> .

`.spec.workloads[].template.spec.volumes[].photonPersistentDisk`

Description

`photonPersistentDisk` represents a `PhotonController` persistent disk attached and mounted on kubelets host machine

Type

`object`

Required

`pdID`

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>pdID</code>	<code>string</code>	<code>pdID</code> is the ID that identifies Photon Controller persistent disk

`.spec.workloads[].template.spec.volumes[].portworxVolume`

Description

portworxVolume represents a portworx volume attached and mounted on kubelets host machine

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

.spec.workloads[].template.spec.volumes[].projected

Description

projected items for all in one resources secrets, configmaps, and downward API

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	<code>defaultMode</code> are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like <code>fsGroup</code>, and the result can be other mode bits set.
<code>sources</code>	<code>array</code>	<code>sources</code> is the list of volume projections. Each entry in this list handles one source.

`.spec.workloads[].template.spec.volumes[].projected.sources`

Description

`sources` is the list of volume projections. Each entry in this list handles one source.

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[]`

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

`object`

Property	Type	Description
<code>clusterTrustBundle</code>	<code>object</code>	ClusterTrustBundle allows a pod to access the <code>.spec.trustBundle</code> field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.
<code>configMap</code>	<code>object</code>	configMap information about the configMap data to project
<code>downwardAPI</code>	<code>object</code>	downwardAPI information about the downwardAPI data to project
<code>secret</code>	<code>object</code>	secret information about the secret data to project

Property	Type	Description
<code>serviceAccountToken</code>	<code>object</code>	serviceAccountToken is information about the serviceAccountToken data to project

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle`

Description

ClusterTrustBundle allows a pod to access the `.spec.trustBundle` field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.`

Type

`object`

Required

`path`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Property	Type	Description
<code>name</code>	<code>string</code>	Select a single ClusterTrustBundle by object name. Mutually-exclusive with <code>signerName</code> and <code>labelSelector</code> .
<code>optional</code>	<code>boolean</code>	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using <code>name</code> , then the named ClusterTrustBundle is allowed not to exist. If using <code>signerName</code> , then the combination of <code>signerName</code> and <code>labelSelector</code> is allowed to match zero ClusterTrustBundles.
<code>path</code>	<code>string</code>	Relative path from the volume root to write the bundle.
<code>signerName</code>	<code>string</code>	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with <code>name</code> . The contents of all selected ClusterTrustBundles will be unified and deduplicated.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector`

Description

Select all ClusterTrustBundles that match this label selector. Only has effect if `signerName` is set. Mutually-exclusive with `name`. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[ ]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]`

Type

string

.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.volumes[].projected.sources[].configMap

Description

configMap information about the configMap data to project

Type

object

Property	Type	Description
<code>items</code>	<code>array</code>	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
<code>optional</code>	<code>boolean</code>	optional specify whether the ConfigMap or its keys must be defined

`.spec.workloads[].template.spec.volumes[].projected.sources[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will

error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
<code>path</code>	<code>string</code>	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI`

Description

downwardAPI information about the downwardAPI data to project

Type

`object`

Property	Type	Description
<code>items</code>	<code>array</code>	Items is a list of DownwardAPIVolume file

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items`

Description

Items is a list of DownwardAPIVolume file

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
<code>fieldRef</code>	object	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
<code>mode</code>	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>path</code>	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
<code>resourceFieldRef</code>	object	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory,

Property	Type	Description
		requests.cpu and requests.memory) are currently supported.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

object

Required

fieldPath

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor		Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret`

Description

secret information about the secret data to project

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified

Property	Type	Description
		paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	optional field specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].projected.sources[].serviceAccountToken`

Description

serviceAccountToken is information about the serviceAccountToken data to project

Type

object

Required

path

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours.Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

.spec.workloads[].template.spec.volumes[].quobyte

Description

quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Type

object

Required

registry

volume

Property	Type	Description
group	string	group to map volume access to Default is no group
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user
volume	string	volume is a string that references an already created Quobyte volume by name.

.spec.workloads[].template.spec.volumes[].rbd

Description

rbd represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/rbd/README.md>

Type

object

Required

image

monitors

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Property	Type	Description
<code>monitors</code>	<code>array</code>	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
<code>pool</code>	<code>string</code>	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
<code>secretRef</code>	<code>object</code>	secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
<code>user</code>	<code>string</code>	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

`.spec.workloads[].template.spec.volumes[].rbd.monitors`

Description

monitors is a collection of Ceph monitors. More info:

<https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

array

`.spec.workloads[].template.spec.volumes[].rbd.monitors[]`

Type

string

`.spec.workloads[].template.spec.volumes[].rbd.secretRef`

Description

secretRef is name of the authentication secret for RBDUser. If provided overrides keyring.

Default is nil. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].scaleIO`

Description

scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Type

`object`

Required

`gateway``secretRef``system`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
<code>gateway</code>	<code>string</code>	gateway is the host address of the ScaleIO API Gateway.
<code>protectionDomain</code>	<code>string</code>	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
<code>readOnly</code>	<code>boolean</code>	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>secretRef</code>	<code>object</code>	secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.
<code>sslEnabled</code>	<code>boolean</code>	sslEnabled Flag enable/disable SSL communication with Gateway, default false

Property	Type	Description
<code>storageMode</code>	<code>string</code>	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
<code>storagePool</code>	<code>string</code>	storagePool is the ScaleIO Storage Pool associated with the protection domain.
<code>system</code>	<code>string</code>	system is the name of the storage system as configured in ScaleIO.
<code>volumeName</code>	<code>string</code>	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

`.spec.workloads[].template.spec.volumes[].scaleIO.secretRef`

Description

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty.

Property	Type	Description
		Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].secret`

Description

secret represents a secret that should populate this volume. More info:
<https://kubernetes.io/docs/concepts/storage/volumes#secret>

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume

Property	Type	Description
		setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>optional</code>	<code>boolean</code>	optional field specify whether the Secret or its keys must be defined
<code>secretName</code>	<code>string</code>	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

`.spec.workloads[].template.spec.volumes[].secret.items`

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

`array`

`.spec.workloads[].template.spec.volumes[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

`object`

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.workloads[].template.spec.volumes[].storageos

Description

storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.

Type

object

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
<code>volumeNamespace</code>	<code>string</code>	<code>volumeNamespace</code> specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set <code>VolumeName</code> to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

`.spec.workloads[].template.spec.volumes[].storageos.secretRef`

Description

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].vsphereVolume`

Description

vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Type

object

Required

`volumePath`

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>storagePolicyID</code>	<code>string</code>	<code>storagePolicyID</code> is the storage Policy Based Management (SPBM) profile ID associated with the <code>StoragePolicyName</code> .
<code>storagePolicyName</code>	<code>string</code>	<code>storagePolicyName</code> is the storage Policy Based Management (SPBM) profile name.
<code>volumePath</code>	<code>string</code>	<code>volumePath</code> is the path that identifies vSphere volume vmdk

.status

Description

`ComponentStatus` defines the observed state of Connector

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	<code>Annotations</code> is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is

Property	Type	Description
		roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See:
<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
status	string	Status of the condition, one of True, False, Unknown.

Property	Type	Description
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits`
 - `DELETE` : delete collection of ConnectorsGit
 - `GET` : list objects of kind ConnectorsGit
 - `POST` : create a new ConnectorsGit
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits/{name}`
 - `DELETE` : delete the specified ConnectorsGit
 - `GET` : read the specified ConnectorsGit
 - `PATCH` : partially update the specified ConnectorsGit
 - `PUT` : replace the specified ConnectorsGit
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits/{name}/status`
 - `GET` : read status of the specified ConnectorsGit
 - `PATCH` : partially update status of the specified ConnectorsGit
 - `PUT` : replace status of the specified ConnectorsGit

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits`

HTTP method

DELETE

Description

delete collection of ConnectorsGit

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ConnectorsGit

HTTP responses

HTTP code	Response body
200 - OK	ConnectorsGitList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ConnectorsGit

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsGit</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGit</code> schema
201 - Created	<code>ConnectorsGit</code> schema
202 - Accepted	<code>ConnectorsGit</code> schema
401 - Unauthorized	Empty

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorgits/{name}

HTTP method

DELETE

Description

delete the specified ConnectorsGit

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ConnectorsGit

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGit</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ConnectorsGit

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGit</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ConnectorsGit

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsGit</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGit</code> schema
201 - Created	<code>ConnectorsGit</code> schema
401 - Unauthorized	Empty

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorgits/{name}/status`

HTTP method

GET

Description

read status of the specified ConnectorsGit

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGit</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ConnectorsGit

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGit</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectorsGit

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsGit</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGit</code> schema
201 - Created	<code>ConnectorsGit</code> schema
401 - Unauthorized	Empty

ConnectorsK8S

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsK8S is the Schema for the ConnectorsK8Ss API.

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsK8SSpec defines the desired state of ConnectorsK8S.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsK8SSpec defines the desired state of ConnectorsK8S.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

.spec.annotations

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

.spec.labels

Description

Labels overrides labels for the namespace and its template.

Type

`object`

.spec.workloads

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template	object	Template describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.spec.workloads[].template

Description

Template describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

Type

object

Property	Type	Description
metadata	ObjectMeta ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
spec	object	Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status ↗

.spec.workloads[].template.spec

Description

Specification of the desired behavior of the pod. More info:

<https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

Type

object

Required

containers

Property	Type	Description
activeDeadlineSeconds	integer	Optional duration in seconds the pod may StartTime before the system will actively tr containers. Value must be a positive intege

Property	Type	Description
<code>affinity</code>	<code>object</code>	If specified, the pod's scheduling constrain
<code>automountServiceAccountToken</code>	<code>boolean</code>	AutomountServiceAccountToken indicates be automatically mounted.
<code>containers</code>	<code>array</code>	List of containers belonging to the pod. Co removed. There must be at least one conta
<code>dnsConfig</code>	<code>object</code>	Specifies the DNS parameters of a pod. Pa to the generated DNS configuration based
<code>dnsPolicy</code>	<code>string</code>	Set DNS policy for the pod. Defaults to "Cl 'ClusterFirstWithHostNet', 'ClusterFirst', 'De in DNSConfig will be merged with the polic DNS options set along with hostNetwork, y to 'ClusterFirstWithHostNet'.
<code>enableServiceLinks</code>	<code>boolean</code>	EnableServiceLinks indicates whether info injected into pod's environment variables, r Optional: Defaults to true.

Property	Type	Description
<code>ephemeralContainers</code>	<code>array</code>	List of ephemeral containers run in this pod. Ephemeral containers complete their life cycle within an existing pod to perform user-initiated actions that require elevated privileges or to perform tasks that cannot be specified when creating a pod, e.g. debugging. In order to add an ephemeral container to an existing pod's ephemeralcontainers subresource.
<code>hostAliases</code>	<code>array</code>	HostAliases is an optional list of hosts and their corresponding IP addresses. Hosts must be present in the /etc/hosts file if specified.
<code>hostIPC</code>	<code>boolean</code>	Use the host's ipc namespace. Optional: Default to false
<code>hostNetwork</code>	<code>boolean</code>	Host networking requested for this pod. Use the host's network namespace. If the option is set to true, the ports that will be used must be specified. Default to false.
<code>hostPID</code>	<code>boolean</code>	Use the host's pid namespace. Optional: Default to false
<code>hostUsers</code>	<code>boolean</code>	Use the host's user namespace. Optional: Default to false. If true, the pod will be run in the host user namespace. This field requires a kernel module with CAP_SYS_MODULE. Only available on linux kernels >= 4.11.0. Setting this field to true will cause the pod's user namespace to be created for the pod. Setting false is useful if you want to use containers that have vulnerabilities even allowing users to run the container with root privileges on the host. This field is only applicable to containers that are created by the kubelet using <code>PodSandbox</code> interface. The kubelet will set this field to true if the <code>PodSandbox</code> interface is used and the <code>HostUserNamespace</code> option is set to true.

Property	Type	Description
hostname	string	Specifies the hostname of the Pod If not set to a system-defined value.
imagePullSecrets	array	ImagePullSecrets is an optional list of references to namespace to use for pulling any of the images. These secrets will be passed to individual pods and used for pulling any of the images. More info: https://kubernetes.io/docs/concepts/configuration/secret#imagepullsecrets-on-a-pod
initContainers	array	List of initialization containers belonging to the pod. Init containers are designed to allow users to run containerized "scripts" before (or alongside) a pod's main containers. They can be used to customize a pod's environment before any containers start. They are intended to be short-lived and are run to completion. If an init container fails, the pod is considered to have failed and is handled accordingly. A pod can be configured to tolerate init container failure (see https://kubernetes.io/docs/concepts/workloads/pods/init-containers/#toleration for more details). Init containers may not have Lifecycle actions, Startup probes, or Readiness probes. The resourceRequirements field is optional for init containers. Limits are applied to init containers in a pod. The podSpecInitContainers field can be used to add or remove init containers. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/init-containers/
nodeName	string	nodeName indicates in which node this pod is to be scheduled. If the field is set, the kubelet for this node becomes a candidate for scheduling the pod. This field should not be used to express an affinity for a specific node. https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodename

Property	Type	Description
nodeSelector	object	NodeSelector is a selector which must be 1 which must match a node's labels for the p info: https://kubernetes.io/docs/concepts/ci
os	object	Specifies the OS of the containers in the p restricted if this is set. If the OS field is set to linux, the following f securityContext.windowsOptions If the OS field is set to windows, following 1 • spec.hostPID • spec.hostIPC • spec.hostUsers • spec.securityContext.appArmorProfile • spec.securityContext.seLinuxOptions • spec.securityContext.seccompProfile • spec.securityContext.fsGroup • spec.securityContext.fsGroupChangeP • spec.securityContext.sysctls • spec.shareProcessNamespace • spec.securityContext.runAsUser • spec.securityContext.runAsGroup • spec.securityContext.supplementalGroi • spec.securityContext.supplementalGroi • spec.containers[*].securityContext.app/ • spec.containers[*].securityContext.seLi • spec.containers[*].securityContext.secc

Property	Type	Description
		- spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readonlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsUser - spec.containers[*].securityContext.runAsGroup
overhead	object	Overhead represents the resource overhead of a Pod given RuntimeClass. This field will be auto populated by the RuntimeClass admission controller. If the FeatureGate is enabled, overhead must not be set in Pod admission controller will reject Pod creation if already set. If RuntimeClass is configured for the pod, overhead will be set to the value defined in the corresponding RuntimeClass. If not set, overhead will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/1141-pod-overhead
preemptionPolicy	string	PreemptionPolicy is the Policy for preempting existing pods. Never, DoNotPreempt, PreemptLowerPriority. Defaults to Never.
priority	integer	The priority value. Various system components use this field to find the priority of the pod. When Priority Admission Controller is enabled, it enforces the pod's priority. The admission controller will not allow a pod to be created if its priority is less than the system's priority. The higher the value, the higher the priority.
priorityClassName	string	If specified, indicates the pod's priority. "system-cluster-critical" and "system-node-critical" are two special keywords which indicate the highest priorities within the system.

Property	Type	Description
		former being the highest priority. Any other PriorityClass object with that name. If not set or zero if there is no default.
readinessGates	array	If specified, all readiness gates will be evaluated when all its containers are ready AND all readiness gates have status equal to "True" More info: https://git.k8s.io/enhancements/keps/sig-network/000001-pod-ready#
resourceClaims	array	ResourceClaims defines which ResourceClaims the Pod needs before the Pod is allowed to start. The resourceClaims list contains the names of the ResourceClaims which consume them by name. This is an alpha field and requires enabling the ResourceClaims feature gate. This field is immutable.
restartPolicy	string	Restart policy for all containers within the pod. Some contexts, only a subset of those values are allowed. More info: https://kubernetes.io/docs/concepts/workloads/pods/#restart-policy
runtimeClassName	string	RuntimeClassName refers to a RuntimeClass object in the K8s API which should be used to run this pod. If no class is named, the pod will not be run. If unspecified, the pod will be run by the default runtime handler. More info: https://kubernetes.io/docs/concepts/workloads/pods/runtime-class/

Property	Type	Description
<code>schedulerName</code>	<code>string</code>	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
<code>schedulingGates</code>	<code>array</code>	SchedulingGates is an opaque list of values that must all be satisfied in order to schedule the pod. If schedulingGates is not empty, the pod will be scheduled by the scheduler only after all gates in this list have been approved by a legitimate authority. If schedulingGates is empty, the pod will be scheduled by the scheduler immediately. SchedulingGates can only be set at pod creation and cannot be updated afterwards.
<code>securityContext</code>	<code>object</code>	SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See the documentation for the fields for each field.
<code>serviceAccount</code>	<code>string</code>	DeprecatedServiceAccount is a deprecated service account to use to run this pod. Deprecated: Use serviceAccountName instead.
<code>serviceAccountName</code>	<code>string</code>	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

Property	Type	Description
<code>setHostnameAsFQDN</code>	<code>boolean</code>	If true the pod's hostname will be configured as the leaf name (the default). In Linux containers, this means setting the registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Fqdn to FQDN. If a pod does not have FQDN, then the pod's hostname will be the same as the node's hostname.
<code>shareProcessNamespace</code>	<code>boolean</code>	Share a single process namespace between all containers in this pod. When this is set to true, every container in the pod is able to view the full process tree of all other containers in the pod, and the first process in the first container is assigned PID 1. HostPID and ShareProcessNamespace cannot both be set to true. Optional: Default to false.
<code>subdomain</code>	<code>string</code>	If specified, the fully qualified Pod hostname will be <code><pod-name>.<subdomain>.<pod-namespace>.svc.cluster.local</code> . If not specified, the pod will not have a domainname at all.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no graceful termination is required). If the value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are allowed to terminate cleanly. The default grace period (on the order of minutes) varies by pod type and is described in the pod's restart policy.
<code>tolerations</code>	<code>array</code>	If specified, the pod's tolerations.

Property	Type	Description
<code>topologySpreadConstraints</code>	<code>array</code>	TopologySpreadConstraints describes how across topology domains. Scheduler will spread the constraints. All topologySpreadConstra
<code>volumes</code>	<code>array</code>	List of volumes that can be mounted by co info: https://kubernetes.io/docs/concepts/st

`.spec.workloads[].template.spec.affinity`

Description

If specified, the pod's scheduling constraints

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	<code>object</code>	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	<code>object</code>	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

.spec.workloads[].template.spec.affinity.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum is (are) the most preferred.

Property	Type	Description
		sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirement specified by this field is not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement specified by this field ceases to be met at some point during pod execution (e.g. due to update), the system may or may not try to eventually evict the pod from its node.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, `requiredDuringScheduling` affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding `matchExpressions`; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
<code>preference</code>	<code>object</code>	A node selector term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

`e.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields`

Description

A list of node selector requirements by node's fields.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key``operator`

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.

Property	Type	Description
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
<code>nodeSelectorTerms</code>	array	Required. A list of node selector terms. The terms are ORed.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelect`

orTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector

orTerms[].matchFields[].values[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding the weight of each element that is matched by the node. The node with the highest sum is the most preferred.

Property	Type	Description
		adding "weight" to the sum if the node has p which matches the corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement specified by this field is not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement specified by this field ceases to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm intersected, i.e. all terms must be satisfied.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod

Property	Type	Description
		labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code> . null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this `PodAffinityTerm` matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

`yTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

`yTerm.namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces`

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector` or

Description

A label query over a set of resources, in this case pods. If it's null, this `PodAffinityTerm` matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchL

abelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace`

`eSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expression specified by this field, it may choose a node that violates one or more the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution anti-affinity expression etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has p

Property	Type	Description
		which matches the corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at the time of scheduling, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm intersected, i.e. all terms must be satisfied.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

Required. A pod affinity term, associated with the corresponding weight.

object

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

`string`

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

finityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values
[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

finityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismat

chLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

paceSelector.matchExpressions[].values[]

Type

string

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paceSelector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces**

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces[]**

Type

string

.spec.workloads[].template.spec.containers

Description

List of containers belonging to the pod. Containers cannot currently be added or removed.
There must be at least one container in a Pod. Cannot be updated.

Type

array

.spec.workloads[].template.spec.containers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image will use this as its command if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must match the container's environment.

Property	Type	Description
		C_IDENTIFIER. All invalid keys will be reported when the container is starting. When a key exists in multiple sources, the value associated with the last source takes precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images field is optional to allow higher level config management to default or override container images in workload Deployments and StatefulSets.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images
lifecycle	object	Actions that the management system should take in response to container lifecycle events. Cannot be updated.
livenessProbe	object	Periodic probe of container liveness. Container v1 will fail if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL).

Property	Type	Description
		Cannot be updated.
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address on the container will be accessible from the network. Must be an array with max length 70. Strategic merge patch may corrupt the data if not specified as an array. See https://github.com/kubernetes/kubernetes/issues/20380 for more info. Cannot be updated.
readinessProbe	object	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
resizePolicy	array	Resources resize policy for the container.
resources	object	Compute Resources required by this container. Cannot be updated. More info: https://kubernetes.io/docs/concepts/configuration/resource-requirements/
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and in that case the only allowed value is "Always". For non-init containers the only allowed value is "Always". If not specified, the restart policy is "Always".

Property	Type	Description
		Pod's restart policy and the container type. Setting RestartPolicy as "Always" for the init container will have the following effect: this init container will be continuously restarted until all regular containers have terminated. Once all regular containers have completed, all init containers with RestartPolicy "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar". Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next container starts immediately after this init container completes, after any startupProbe has successfully completed.
<code>securityContext</code>	<code>object</code>	SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/
<code>startupProbe</code>	<code>object</code>	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully completes. If this probe fails, the Pod will be restarted. This can be used to probe parameters at the beginning of a Pod's life cycle. The probe might take a long time to load data or warm a cache before entering steady-state operation. This cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data at container runtime. If this is not set, reads from stdin will return EOF.

Property	Type	Description
		container will always result in EOF. Default is false.
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When false, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process from stdin will never receive an EOF. Default is false.
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container termination message will be written is mounted in the container's filesystem. Message written is intended to be of final status, such as an assertion failure message. Message length across all containers will be limited by the node if greater than 4096 bytes. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. If set to File, will use the contents of terminationMessagePath. If set to FallbackToLogsOnError, container status message on both success and failure will use the contents of terminationMessagePath if available, and will fallback to the container log output if the termination message file is empty or exited with an error. The log output is limited to 2048 bytes, whichever is smaller. Defaults to File. Cannot be updated.

Property	Type	Description
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be u container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystems updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be co container image. Cannot be updated.

`.spec.workloads[].template.spec.containers[].args`

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$ (VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

`array`

`.spec.workloads[].template.spec.containers[].args[]`

Type

string

`.spec.workloads[].template.spec.containers[].command`

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$$(VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.containers[].command[]`

Type

string

`.spec.workloads[].template.spec.containers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.containers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

object

Property	Type	Description
<code>configMapKeyRef</code>	object	Selects a key of a ConfigMap.
<code>fieldRef</code>	object	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
<code>resourceFieldRef</code>	object	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	object	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.containers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

object

Required

key

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap or its key must be defined

.spec.workloads[].template.spec.containers[].env[].valueFrom.fieldRef

Description

Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Type

object

Required

fieldPath

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.containers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>string</code>	Specifies the output format of the exposed resources, defaults to "1"

Property	Type	Description
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.containers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key of the secret to select from. Must be a valid secret key.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.containers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[].envFrom[]`

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	<code>object</code>	The ConfigMap to select from
<code>prefix</code>	<code>string</code>	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	<code>object</code>	The Secret to select from

`.spec.workloads[].template.spec.containers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap must be defined

.spec.workloads[].template.spec.containers[].envFrom[].secretRef

Description

The Secret to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.containers[].lifecycle`

Description

Actions that the management system should take in response to container lifecycle events.
Cannot be updated.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
<code>preStop</code>	<code>object</code>	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the

Property	Type	Description
		container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗

`.spec.workloads[].template.spec.containers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>string</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name``value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perform.
<code>sleep</code>	<code>object</code>	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	<code>object</code>	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.tcpSocket`

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].livenessProbe`

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies an action involving a GRP
<code>httpGet</code>	object	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	integer	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
<code>periodSeconds</code>	integer	How often (in seconds) to perform the prot 10 seconds. Minimum value is 1.
<code>successThreshold</code>	integer	Minimum consecutive successes for the pr considered successful after having failed. I

Property	Type	Description
		Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocket specifies an action involving a
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to gracefully shutdown before it is terminated. The grace period is the duration in seconds after the processes running on the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It specifies that the container managed liveness probes of the pod must not fail after this duration. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via a kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes

.spec.workloads[].template.spec.containers[].livenessProbe.exec

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.workloads[].template.spec.containers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].livenessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].ports`

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

object

Required

containerPort

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.containers[].readinessProbe`

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period in seconds after the processes running on the pod are sent a termination signal and the time until the processes are forcibly halted with a kill signal. A value greater than 0 indicates the grace period should be used. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/probe-lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].readinessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.containers[].readinessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.workloads[].template.spec.containers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.containers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.containers[].resources`

Description

Compute Resources required by this container. Cannot be updated. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.containers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.containers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.containers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.containers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.containers[].securityContext`

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
<code>procMount</code>	<code>string</code>	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	object	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.containers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if <code>type</code> is "Localhost".
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.containers[].securityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.containers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.containers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.containers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.containers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running on the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.containers[].startupProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.containers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.containers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.containers[].startupProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].volumeDevices`

Description

volumeDevices is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.containers[].volumeDevices[]`

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath`

`name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	devicePath is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.containers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.containers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain '..'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.dnsConfig`

Description

Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy.

Type

`object`

Property	Type	Description
<code>nameservers</code>	<code>array</code>	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
<code>options</code>	<code>array</code>	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.
<code>searches</code>	<code>array</code>	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

`.spec.workloads[].template.spec.dnsConfig.nameservers`

Description

A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.

Type

`array`

`.spec.workloads[].template.spec.dnsConfig.nameservers[]`

Type

`string`

`.spec.workloads[].template.spec.dnsConfig.options`

Description

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type

array

`.spec.workloads[].template.spec.dnsConfig.options[]`

Description

PodDNSConfigOption defines DNS resolver options of a pod.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Required.
<code>value</code>	<code>string</code>	

`.spec.workloads[].template.spec.dnsConfig.searches`

Description

A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

Type

array

`.spec.workloads[].template.spec.dnsConfig.searches[]`

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers`

Description

List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[]`

Description

An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.

Type

`object`

Required

`name`

Property	Type	Description
<code>args</code>	<code>array</code>	Arguments to the entrypoint. The image's CMD is not provided. Variable references \$(VAR_NAME) using the container's environment. If a variable c

Property	Type	Description
		resolved, the reference in the input string will be expanded. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
<code>command</code>	<code>array</code>	Entrypoint array. Not executed within a shell. The ENTRYPOINT is used if this is not provided. Variables referenced with \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
<code>env</code>	<code>array</code>	List of environment variables to set in the container. Cannot be updated.

Property	Type	Description
<code>envFrom</code>	<code>array</code>	List of sources to populate environment variables in the container. The keys defined within a source must match the C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source with the same key takes precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
<code>image</code>	<code>string</code>	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images
<code>imagePullPolicy</code>	<code>string</code>	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images
<code>lifecycle</code>	<code>object</code>	Lifecycle is not allowed for ephemeral containers
<code>livenessProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>name</code>	<code>string</code>	Name of the ephemeral container specified as a pod's spec. This name must be unique among all containers, and ephemeral containers.

Property	Type	Description
<code>ports</code>	<code>array</code>	Ports are not allowed for ephemeral containers.
<code>readinessProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	Resources are not allowed for ephemeral containers use spare resources already allocated
<code>restartPolicy</code>	<code>string</code>	Restart policy for the container to manage the restart of each container within a pod. This may only be set on non-ephemeral containers. You cannot set this field on ephemeral containers.
<code>securityContext</code>	<code>object</code>	Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the SecurityContext override the equivalent fields of PodSecurityContext.
<code>startupProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data at container runtime. If this is not set, reads from stdin will always result in EOF. Default is false

Property	Type	Description
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When false, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on pod start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the pod is restarted. If this flag is false, a container process starting from stdin will never receive an EOF. Default is false.
<code>targetContainerName</code>	<code>string</code>	If set, the name of the container from PodSpec that this ephemeral container targets. The ephemeral container will run in the namespaces (IPC, PID, etc) of this container. If not set, then the ephemeral container uses the namespaces of the Pod spec. The container runtime must implement support for namespace targeting. If the runtime does not support namespace targeting, the result of setting this field is undefined.
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container's termination message will be written is mounted in the container's filesystem. Message written is intended to be consumed by a user-defined process or another instance of the container image used for this container. Must be a relative path of the container image. Message length across all containers will be limited by the node. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. If set to truncate, will use the contents of terminationMessagePath and truncate it to the configured message length.

Property	Type	Description
		container status message on both success and failure. FallbackToLogsOnError will use the last chunk of output if the termination message file is empty and exited with an error. The log output is limited to 2048 lines, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for stdin. Requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be mapped into the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be configurable by the container image. Cannot be updated.

`.spec.workloads[].template.spec.ephemeralContainers[].args`

Description

Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `$(foo $(foo $(bar)))`.

"\$\$ (VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].args[]

Type

string

.spec.workloads[].template.spec.ephemeralContainers[].command

Description

Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].command[]

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. Must be a C_IDENTIFIER.
<code>value</code>	<code>string</code>	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME)

Property	Type	Description
		syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

object

Required

resource

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	☐	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
<code>key</code>	<code>string</code>	The key of the secret to select from. Must be a valid secret key.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[]`

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	object	The ConfigMap to select from
<code>prefix</code>	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	object	The Secret to select from

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[].secretRef`

Description

The Secret to select from

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle`

Description

Lifecycle is not allowed for ephemeral containers.

Type

object

Property	Type	Description
<code>postStart</code>	object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks
<code>preStop</code>	object	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].ifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].ifecycle.postStart.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be

Property	Type	Description
		an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility.

Property	Type	Description
		There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec.command[]

Type

string

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.

Property	Type	Description
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when

tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u:
timeoutSeconds	integer	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. <a href="https://kubernetes.io/docs/concepts/worklo
lifecycle#container-probes">https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`**Required**`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type`object`**Required**`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].li venessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].ports`

Description

Ports are not allowed for ephemeral containers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe`

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-life-cycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.ephemeralContainers[].resources`

Description

Resources are not allowed for ephemeral containers. Ephemeral containers use spare resources already allocated to the pod.

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.ephemeralContainers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.ephemeralContainers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.ephemeralContainers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext`

Description

Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe`

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.
failureThreshold	integer	Minimum consecutive failures for the probe considered failed after having succeeded.

Property	Type	Description
		Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRP
httpGet	object	HTTPGet specifies the http request to perf
initialDelaySeconds	integer	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
periodSeconds	integer	How often (in seconds) to perform the prot 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
tcpSocket	object	TCPSocket specifies an action involving a
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim

Property	Type	Description
		process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to specify the shell in the container's image.

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>string</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name``value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].volumeDevices[]`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.ephemeralContainers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.hostAliases`

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

`array`

`.spec.workloads[].template.spec.hostAliases[]`

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type

object

Required

ip

Property	Type	Description
hostnames	array	Hostnames for the above IP address.
ip	string	IP address of the host file entry.

.spec.workloads[].template.spec.hostAliases[].hostnames**Description**

Hostnames for the above IP address.

Type

array

**.spec.workloads[].template.spec.hostAliases[].hostnames
[]****Type**

string

.spec.workloads[].template.spec.imagePullSecrets**Description**

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info:

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

Type

array

`.spec.workloads[].template.spec.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.initContainers`

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init

containers cannot currently be added or removed. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

Type

array

```
.spec.workloads[].template.spec.initContainers[]
```

Description

A single application container that you want to run within a pod.

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be resolved, the reference in the input string will be expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be expanded using the container's environment. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" to produce a string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be expanded using the container's environment. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" to produce a string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

Property	Type	Description
		the input string will be unchanged. Double \$\$ are single \$, which allows for escaping the \$(VAR_NAME) i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Can be updated. More info: https://kubernetes.io/docs/tasks/configure-pod-container/define-command-argument-container-command-in-a-shell/
env	array	List of environment variables to set in the container. Can be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must match the C_IDENTIFIER. All invalid keys will be reported as an error when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images/ field is optional to allow higher level config management to default or override container images in workload Deployments and StatefulSets.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images/#image-pull-policy

Property	Type	Description
		https://kubernetes.io/docs/concepts/containers/images ↗
lifecycle	object	Actions that the management system should take in response to container lifecycle events. Cannot be updated.
livenessProbe	object	Periodic probe of container liveness. Container v1.4.0+ has a default liveness probe that fails if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. You must specify a port to be able to access the service. If a container is listening on the default "0.0.0.0" address, this port will be accessible from the network. Maximum 100 ports per container. This array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/20380 Cannot be updated.
readinessProbe	object	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

Property	Type	Description
		https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	Compute Resources required by this container. (Can be updated. More info: https://kubernetes.io/docs/concepts/configuration/resource-limits)
<code>restartPolicy</code>	<code>string</code>	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers, this field is not specified, the restart behavior is controlled by the Pod's restart policy and the container type. Setting RestartPolicy to "Always" for the init container will have the following effect: this init container will be continually restarted until all regular containers have terminated. Once all regular containers have completed, all init containers with "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar". Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container after any startupProbe has successfully completed.
<code>securityContext</code>	<code>object</code>	SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/concepts/security/pod-security/

Property	Type	Description
		https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ ↗
<code>startupProbe</code>	<code>object</code>	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully. If this probe fails, the Pod will be restarted. This can be used to probe parameters at the beginning of a Pod's life cycle. This cannot be updated. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data in its container runtime. If this is not set, reads from stdin will always result in EOF. Default is false.
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When set to true, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on pod start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the pod is restarted. If this flag is false, a container process from stdin will never receive an EOF. Default is false.
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container's termination message will be written is mounted in the container's filesystem. Message written is intended to be consumed by a downstream component. Default path is /dev/termination-log, which requires mount access for the container and is only available on Kubernetes 1.5 and above.

Property	Type	Description
		final status, such as an assertion failure message truncated by the node if greater than 4096 bytes. message length across all containers will be limited. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. Will use the contents of terminationMessagePath on container status message on both success and failure. FallbackToLogsOnError will use the last chunk of output if the termination message file is empty and exited with an error. The log output is limited to 2 lines, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for stdin. Requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used within the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be configurable on the container image. Cannot be updated.

`.spec.workloads[].template.spec.initContainers[].args`

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$$(VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.initContainers[].args[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].command`

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$$(VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.initContainers[].command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.initContainers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. Must be a C_IDENTIFIER.
<code>value</code>	<code>string</code>	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input

Property	Type	Description
		string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

object

Required

resource

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	☐	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.initContainers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.initContainers[].envFrom`



Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	object	The ConfigMap to select from
<code>prefix</code>	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	object	The Secret to select from

`.spec.workloads[].template.spec.initContainers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
<code>name</code>	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap must be defined

`.spec.workloads[].template.spec.initContainers[].envFrom[].secretRef`

Description

The Secret to select from

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.initContainers[].lifecycle`

Description

Actions that the management system should take in response to container lifecycle events.

Cannot be updated.

Type

object

Property	Type	Description
postStart	object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks
preStop	object	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the

container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.
httpGet	object	HTTPGet specifies the http request to perform.
sleep	object	Sleep represents the duration that the container should sleep before being terminated.
tcpSocket	object	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

**.spec.workloads[].template.spec.initContainers[].lifecycle.
postStart.httpGet.httpHeaders**

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.

Property	Type	Description
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when

tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.workloads[].template.spec.initContainers[].livenessProbe

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sigi value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u:
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. <a href="https://kubernetes.io/docs/concepts/worklo
lifecycle#container-probes">https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`**Required**`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type`object`**Required**`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].ports`

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.initContainers[].readinessProbe`

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.initContainers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].resizePolicy`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.initContainers[].resource` **S**

Description

Compute Resources required by this container. Cannot be updated. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.initContainers[].resource` `s.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].resource` `s.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.initContainers[].resource` `s.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.initContainers[].resource` `s.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.initContainers[].security`

Context

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	object	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.initContainers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.initContainers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.initContainers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.initContainers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.initContainers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the prok 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period duration in seconds after the processes running in the pod are sent a termination signal and the time until the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].volumeDevice`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.initContainers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.nodeSelector`

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: <https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

`object`

`.spec.workloads[].template.spec.os`

Description

Specifies the OS of the containers in the pod. Some pod and container fields are restricted if this is set. If the OS field is set to linux, the following fields must be unset: - securityContext.windowsOptions If the OS field is set to windows, following fields must be unset: - spec.hostPID - spec.hostIPC - spec.hostUsers - spec.securityContext.appArmorProfile - spec.securityContext.seLinuxOptions - spec.securityContext.seccompProfile - spec.securityContext.fsGroup - spec.securityContext.fsGroupChangePolicy - spec.securityContext.sysctls - spec.shareProcessNamespace - spec.securityContext.runAsUser - spec.securityContext.runAsGroup - spec.securityContext.supplementalGroups - spec.securityContext.supplementalGroupsPolicy - spec.containers[*].securityContext.appArmorProfile - spec.containers[*].securityContext.seLinuxOptions - spec.containers[*].securityContext.seccompProfile - spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readOnlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsUser - spec.containers[*].securityContext.runAsGroup

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration ↗ Clients should expect to handle additional values and treat unrecognized values in this field as os: null

`.spec.workloads[].template.spec.overhead`

Description

Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info:

<https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md>

Type

object

`.spec.workloads[].template.spec.readinessGates`

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type

array

`.spec.workloads[].template.spec.readinessGates[]`

Description

PodReadinessGate contains the reference to a pod condition

Type

object

Required

conditionType

Property	Type	Description
<code>conditionType</code>	<code>string</code>	ConditionType refers to a condition in the pod's condition list with matching type.

`.spec.workloads[].template.spec.resourceClaims`

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.

Type

`array`

`.spec.workloads[].template.spec.resourceClaims[]`

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod. It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name uniquely identifies this resource claim inside the pod. This must be a

Property	Type	Description
		DNS_LABEL.
resourceClaimName	string	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.
resourceClaimTemplateName	string	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in pod.status.resourceClaimStatuses. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

`.spec.workloads[].template.spec.schedulingGates`

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

array

`.spec.workloads[].template.spec.schedulingGates[]`

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the scheduling gate. Each scheduling gate must have a unique name field.

`.spec.workloads[].template.spec.securityContext`

Description

SecurityContext holds pod-level security attributes and common container settings.
Optional: Defaults to empty. See type description for default values of each field.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>fsGroup</code>	<code>integer</code>	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>fsGroupChangePolicy</code>	<code>string</code>	<code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and

Property	Type	Description
		"Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that

Property	Type	Description
		container. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
supplementalGroups	array	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the

Property	Type	Description
		supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
supplementalGroupsPolicy	string	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
sysctls	array	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	object	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.securityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext.

If set in both `SecurityContext` and `PodSecurityContext`, the value specified in `SecurityContext` takes precedence for that container. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

object

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.securityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.securityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

`array`

`.spec.workloads[].template.spec.securityContext.supplementalGroups[]`

Type

integer

`.spec.workloads[].template.spec.securityContext.sysctls`

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

`.spec.workloads[].template.spec.securityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.workloads[].template.spec.securityContext.windows

Options

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	<code>GMSACredentialSpecName</code> is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	<code>HostProcess</code> determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective <code>HostProcess</code> value (it is not allowed to have a mix of <code>HostProcess</code> containers and non- <code>HostProcess</code> containers). In addition, if <code>HostProcess</code> is true then <code>HostNetwork</code> must also be set to true.

Property	Type	Description
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.tolerations`

Description

If specified, the pod's tolerations.

Type

`array`

`.spec.workloads[].template.spec.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.workloads[].template.spec.topologySpreadConstraints`

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[]`

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

object

Required

maxSkew

topologyKey

whenUnsatisfiable

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values

Property	Type	Description
		from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).
maxSkew	integer	MaxSkew describes the degree to which pods may be unevenly distributed. When <code>whenUnsatisfiable=DoNotSchedule</code>, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1).

Property	Type	Description
		if MaxSkew is 2, incoming pod can be scheduled onto any zone. When <code>whenUnsatisfiable=ScheduleAnyway</code> , it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.

Property	Type	Description
<code>minDomains</code>	<code>integer</code>	MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule. For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: zone1 zone2 zone3 P P P P P P The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.
<code>nodeAffinityPolicy</code>	<code>string</code>	NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are:

Property	Type	Description
		- Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. - Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations. If this value is nil, the behavior is equivalent to the Honor policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
nodeTaintsPolicy	string	NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: - Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. - Ignore: node taints are ignored. All nodes are included. If this value is nil, the behavior is equivalent to the Ignore policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
topologyKey	string	TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket.

Property	Type	Description
		We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of <code>nodeAffinityPolicy</code> and <code>nodeTaintsPolicy</code>. e.g. If <code>TopologyKey</code> is "kubernetes.io/hostname", each <code>Node</code> is a domain of that topology. And, if <code>TopologyKey</code> is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.
<code>whenUnsatisfiable</code>	<code>string</code>	<code>WhenUnsatisfiable</code> indicates how to deal with a pod if it doesn't satisfy the spread constraint. • <code>DoNotSchedule</code> (default) tells the scheduler not to schedule it. • <code>ScheduleAnyway</code> tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same <code>labelSelector</code> spread as 3/1/1: zone1 zone2 zone3 P P P P P If <code>WhenUnsatisfiable</code> is set to <code>DoNotSchedule</code>, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as <code>ActualSkew(2-1)</code> on zone2(zone3) satisfies <code>MaxSkew(1)</code>. In other words, the cluster can still be imbalanced, but scheduler won't make it <i>more</i> imbalanced. It's a required field.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector`

Description

LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.topologySpreadConstraints[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].matchLabelKeys[]`

Type

`string`

`.spec.workloads[].template.spec.volumes`

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

`array`

`.spec.workloads[].template.spec.volumes[]`

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

`object`

Required

`name`

Property	Type	Description
<code>awsElasticBlockStore</code>	<code>object</code>	<code>awsElasticBlockStore</code> represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. https://kubernetes.io/docs/concepts/storage/volumes/#aws-elastic-block-store

Property	Type	Description
<code>azureDisk</code>	<code>object</code>	azureDisk represents an Azure Data Disk mount on mount to the pod.
<code>azureFile</code>	<code>object</code>	azureFile represents an Azure File Service mount or mount to the pod.
<code>cephfs</code>	<code>object</code>	cephFS represents a Ceph FS mount on the host th lifetime
<code>cinder</code>	<code>object</code>	cinder represents a cinder volume attached and mot machine. More info: https://examples.k8s.io/mysql-c
<code>configMap</code>	<code>object</code>	configMap represents a configMap that should popu
<code>csi</code>	<code>object</code>	csi (Container Storage Interface) represents ephem handled by certain external CSI drivers (Beta feature
<code>downwardAPI</code>	<code>object</code>	downwardAPI represents downward API about the p populate this volume

Property	Type	Description
<code>emptyDir</code>	<code>object</code>	<code>emptyDir</code> represents a temporary directory that shares its life cycle with the pod (the volume is present as long as the pod is present and is deleted when the pod is deleted or restarts). More info: https://kubernetes.io/docs/concepts/storage/emptydir
<code>ephemeral</code>	<code>object</code>	<code>ephemeral</code> represents a volume that is handled by a <code>EphemeralVolumeSource</code>. The volume's lifecycle is tied to the pod that defines it (the volume is present as long as the pod is present and is deleted when the pod is deleted or restarts). Use this if: a) the volume is only needed while the pod is running, b) the pod needs normal volumes like restoring from snapshot or capacity, c) the storage driver is specified through a <code>storageClassName</code> and the storage driver supports dynamic volume provisioning. Use <code>PersistentVolumeClaim</code> or one of the vendor-specific <code>StorageClasses</code> for volumes that persist for longer than the lifecycle of a pod. Use CSI for light-weight local ephemeral volumes if it is not meant to be used that way - see the documentation for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.
<code>fc</code>	<code>object</code>	<code>fc</code> represents a Fibre Channel resource that is attached to a storage machine and then exposed to the pod.

Property	Type	Description
<code>flexVolume</code>	<code>object</code>	<code>flexVolume</code> represents a generic volume resource that is provisioned/attached using an exec based plugin.
<code>flocker</code>	<code>object</code>	<code>flocker</code> represents a Flocker volume attached to a Kubernetes node. This depends on the Flocker control service being running on the node.
<code>gcePersistentDisk</code>	<code>object</code>	<code>gcePersistentDisk</code> represents a GCE Disk resource that is provisioned on the kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes/#gce-persistent-disk
<code>gitRepo</code>	<code>object</code>	<code>gitRepo</code> represents a git repository at a particular revision. <code>GitRepo</code> is deprecated. To provision a container with a git repo, use an <code>EmptyDir</code> into an <code>InitContainer</code> that clones the repo and then expose the <code>EmptyDir</code> into the Pod's container.
<code>glusterfs</code>	<code>object</code>	<code>glusterfs</code> represents a Glusterfs mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/glusterfs/
<code>hostPath</code>	<code>object</code>	<code>hostPath</code> represents a pre-existing file or directory on the host that is directly exposed to the container. This is generally used for logging agents or other privileged things that are allowed to see the host's file system. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes/#host-path

Property	Type	Description
image	object	image represents an OCI object (a container image mounted on the kubelet's host machine. The volume startup depending on which PullPolicy value is provided • Always: the kubelet always attempts to pull the reference, even if it already exists. Container creation will fail if the pull fails. • Never: the kubelet never pulls the reference and only uses the image or artifact if it is already present on disk. Container creation will fail if the reference is not present. • IfNotPresent: the kubelet pulls if the reference is not present on disk. Container creation will fail if the reference is not present and pull fails. The volume gets re-resolved if the pod gets deleted which means that new remote content will become available. Failure to resolve or pull the image during pod startup from starting and may add significant latency. Failure to pull the image will be reported on the pod status message. The types of objects that may be mounted are defined by the container runtime implementation on the host. The minimum must include all valid types supported by the container runtime. The OCI object gets mounted in a single direct layer (spec.containers[].volumeMounts.mountPath) by the container runtime in the same way as for container images. The volume is mounted read-only (ro) and non-executable files (noexec) are not supported for containers are not supported (spec.containers[].volumeMounts.subpath). The field spec.securityContext.fsGroupChangePolicy has no effect on this type.

Property	Type	Description
iscsi	object	iscsi represents an ISCSI Disk resource that is attached to the host machine and then exposed to the pod. More info: https://examples.k8s.io/volumes/iscsi/README.md
name	string	name of the volume. Must be a DNS_LABEL and unique within the namespace. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	object	nfs represents an NFS mount on the host that shares a pod-readable directory. More info: https://kubernetes.io/docs/concepts/storage/volumes/#nfs
persistentVolumeClaim	object	persistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes/#persistentvolumeclaims
photonPersistentDisk	object	photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine
portworxVolume	object	portworxVolume represents a portworx volume attached and mounted on kubelets host machine
projected	object	projected items for all in one resources secrets, configmaps, downward API

Property	Type	Description
<code>quobyte</code>	<code>object</code>	quobyte represents a Quobyte mount on the host throughout the pod's lifetime
<code>rbd</code>	<code>object</code>	rbd represents a Rados Block Device mount on the host throughout the pod's lifetime. More info: https://examples.k8s.io/volumes/rbd/README.md
<code>scaleIO</code>	<code>object</code>	scaleIO represents a ScaleIO persistent volume attached to the pod. More info: https://examples.k8s.io/volumes/scaleio/README.md
<code>secret</code>	<code>object</code>	secret represents a secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret
<code>storageos</code>	<code>object</code>	storageOS represents a StorageOS volume attached to the pod. More info: https://examples.k8s.io/volumes/storageos/README.md
<code>vsphereVolume</code>	<code>object</code>	vsphereVolume represents a vSphere volume attached to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#vspherevolume

`.spec.workloads[].template.spec.volumes[].awsElasticBlockStore`

Description

awsElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
partition	integer	partition is the partition in the volume that you want to mount. If not specified, the default is to mount by volume name. Examples: For volume "/dev/sda1", you specify the partition as "1". Similarly, the volume partition for "/dev/sda" is "0" (or you can leave the property empty).
readOnly	boolean	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

`.spec.workloads[].template.spec.volumes[].azureDisk`

Description

azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

object

Required

diskName

diskURI

Property	Type	Description
<code>cachingMode</code>	<code>string</code>	cachingMode is the Host Caching mode: None, Read Only, Read Write.
<code>diskName</code>	<code>string</code>	diskName is the Name of the data disk in the blob storage
<code>diskURI</code>	<code>string</code>	diskURI is the URI of data disk in the blob storage
<code>fsType</code>	<code>string</code>	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>kind</code>	<code>string</code>	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> Defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .

`.spec.workloads[].template.spec.volumes[].azureFile`

Description

`azureFile` represents an Azure File Service mount on the host and bind mount to the pod.

Type

`object`

Required

`secretName`

`shareName`

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .
<code>secretName</code>	<code>string</code>	<code>secretName</code> is the name of secret that contains Azure Storage Account Name and Key
<code>shareName</code>	<code>string</code>	<code>shareName</code> is the azure share Name

`.spec.workloads[].template.spec.volumes[].cephfs`

Description

`cephFS` represents a Ceph FS mount on the host that shares a pod's lifetime

Type

object

Required

monitors

Property	Type	Description
<code>monitors</code>	<code>array</code>	<code>monitors</code> is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>path</code>	<code>string</code>	<code>path</code> is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>secretFile</code>	<code>string</code>	<code>secretFile</code> is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

Property	Type	Description
<code>secretRef</code>	<code>object</code>	secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>user</code>	<code>string</code>	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

`.spec.workloads[].template.spec.volumes[].cephfs.monitors`

Description

monitors is Required: Monitors is a collection of Ceph monitors More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

`array`

`.spec.workloads[].template.spec.volumes[].cephfs.monitors[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].cephfs.secretRef`

Description

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.workloads[].template.spec.volumes[].cinder

Description

cinder represents a cinder volume attached and mounted on kubelets host machine. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

Type

object

Required

volumeID

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code>. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is optional: points to a secret object containing parameters used to connect to OpenStack.
<code>volumeID</code>	<code>string</code>	<code>volumeID</code> used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

`.spec.workloads[].template.spec.volumes[].cinder.secretRef`

Description

`secretRef` is optional: points to a secret object containing parameters used to connect to OpenStack.

Type

`object`

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].configMap`

Description

configMap represents a configMap that should populate this volume

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified

Property	Type	Description
		paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

`.spec.workloads[].template.spec.volumes[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].csi`

Description

csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Type

object

Required

driver

Property	Type	Description
driver	string	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
fsType	string	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
nodePublishSecretRef	object	nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.
readOnly	boolean	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).

Property	Type	Description
<code>volumeAttributes</code>	<code>object</code>	<code>volumeAttributes</code> stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

`.spec.workloads[].template.spec.volumes[].csi.nodePublishSecretRef`

Description

`nodePublishSecretRef` is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI `NodePublishVolume` and `NodeUnpublishVolume` calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.spec.workloads[].template.spec.volumes[].csi.volumeAttributes`

Description

volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

object

`.spec.workloads[].template.spec.volumes[].downwardAPI`

Description

downwardAPI represents downward API about the pod that should populate this volume

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	Items is a list of downward API volume file

`.spec.workloads[].template.spec.volumes[].downwardAPI.items`

Description

Items is a list of downward API volume file

Type

array

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
<code>fieldRef</code>	<code>object</code>	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
<code>mode</code>	<code>integer</code>	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
<code>path</code>	<code>string</code>	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[].resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>string</code>	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.volumes[].emptyDir`

Description

emptyDir represents a temporary directory that shares a pod's lifetime. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#emptydir>

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
sizeLimit		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.workloads[].template.spec.volumes[].ephemeral

Description

ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see

EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual pod. Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.

Type

object

Property	Type	Description
<code>volumeClaimTemplate</code>	object	Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be <code><pod name>-<volume name></code> where <code><volume name></code> is the name from the <code>PodSpec.Volumes</code> array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will <i>not</i> be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created.

Property	Type	Description
		Required, must not be nil.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate`

Description

Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be `<pod name>-<volume name>` where `<volume name>` is the name from the `PodSpec.Volumes` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

Type

object

Required

spec

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code>	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.

Property	Type	Description
<code>spec</code>	<code>object</code>	The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec`

Description

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

Type

`object`

Property	Type	Description
<code>accessModes</code>	<code>array</code>	accessModes contains the desired access mode volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
<code>dataSource</code>	<code>object</code>	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can support the specified data source, it will create a new

Property	Type	Description
		volume based on the contents of the specific source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If dataSourceRef.namespace is specified, then dataSourceRef contents will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-core API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the name of the specified object matches some installed volume populator or dynamic provisioner. This field will take precedence over the dataSource field and as long as both fields are non-empty, they must have the same value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: • While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves &

Property	Type	Description
		values, and generates an error if a disallowe value is specified. While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
resources	object	resources represents the minimum resources th volume should have. If RecoverVolumeExpansionFailure feature is ena users are allowed to specify resource requirem that are lower than previous value but must still higher than capacity recorded in the status field claim. More info: https://kubernetes.io/docs/concepts/storage/per-volumes#resources
selector	object	selector is a label query over volumes to consid binding.
storageClassName	string	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per-volumes#class-1

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that the <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string value is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClassName</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ [↗] (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]`

Type

string

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef`

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.

(Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:
<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector`

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.workloads[].template.spec.volumes[].fc`

Description

fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Type

`object`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>lun</code>	<code>integer</code>	lun is Optional: FC target lun number
<code>readOnly</code>	<code>boolean</code>	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in

Property	Type	Description
		VolumeMounts.
<code>targetWWNs</code>	<code>array</code>	targetWWNs is Optional: FC target worldwide names (WWNs)
<code>wwids</code>	<code>array</code>	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

.spec.workloads[].template.spec.volumes[].fc.targetWWNs

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

`array`

.spec.workloads[].template.spec.volumes[].fc.targetWWNs
[]

Type

`string`

.spec.workloads[].template.spec.volumes[].fc.wwids

Description

wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

Type

`array`

`.spec.workloads[].template.spec.volumes[].fc.wwid[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].flexVolume`

Description

flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type

`object`

Required

`driver`

Property	Type	Description
<code>driver</code>	<code>string</code>	driver is the name of the driver to use for this volume.
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
<code>options</code>	<code>object</code>	options is Optional: this field holds extra command options if any.

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>secretRef</code>	<code>object</code>	secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

`.spec.workloads[].template.spec.volumes[].flexVolume.options`

Description

options is Optional: this field holds extra command options if any.

Type

`object`

`.spec.workloads[].template.spec.volumes[].flexVolume.secretRef`

Description

secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified.
If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].flocker`

Description

flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Type

`object`

Property	Type	Description
<code>datasetName</code>	<code>string</code>	datasetName is Name of the dataset stored as metadata - > name on the dataset for Flocker should be considered as deprecated
<code>datasetUUID</code>	<code>string</code>	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

`.spec.workloads[].template.spec.volumes[].gcePersistentDisk`

Description

`gcePersistentDisk` represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

Type

object

Required

pdName

Property	Type	Description
<code>fsType</code>	string	<code>fsType</code> is filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
<code>partition</code>	integer	<code>partition</code> is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the partition for /dev/sda is "0" (or you can leave the property empty) if you want to select the whole disk. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
<code>pdName</code>	string	<code>pdName</code> is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis

`.spec.workloads[].template.spec.volumes[].gitRepo`

Description

gitRepo represents a git repository at a particular revision. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
<code>directory</code>	<code>string</code>	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
<code>repository</code>	<code>string</code>	repository is the URL

Property	Type	Description
<code>revision</code>	<code>string</code>	revision is the commit hash for the specified revision.

`.spec.workloads[].template.spec.volumes[].glusterfs`

Description

glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/glusterfs/README.md>

Type

`object`

Required

`endpoints`

`path`

Property	Type	Description
<code>endpoints</code>	<code>string</code>	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
<code>path</code>	<code>string</code>	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

`.spec.workloads[].template.spec.volumes[].hostPath`

Description

hostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

`.spec.workloads[].template.spec.volumes[].image`

Description

image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided:

- Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails.
- Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present.
- IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (`spec.containers[*].volumeMounts.mountPath`) by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (`spec.containers[*].volumeMounts.subpath`). The field `spec.securityContext.fsGroupChangePolicy` has no effect on this volume type.

Type

object

Property	Type	Description
<code>pullPolicy</code>	<code>string</code>	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.
<code>reference</code>	<code>string</code>	Required: Image or artifact reference to be used. Behaves in the same way as <code>pod.spec.containers[*].image</code>. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

`.spec.workloads[].template.spec.volumes[].iscsi`

Description

iscsi represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://examples.k8s.io/volumes/iscsi/README.md>

Type

`object`

Required

`iqn`

`lun`

`targetPortal`

Property	Type	Description
<code>chapAuthDiscovery</code>	<code>boolean</code>	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
<code>chapAuthSession</code>	<code>boolean</code>	chapAuthSession defines whether support iSCSI Session CHAP authentication
<code>fsType</code>	<code>string</code>	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
<code>initiatorName</code>	<code>string</code>	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
<code>iqn</code>	<code>string</code>	iqn is the target iSCSI Qualified Name.
<code>iscsiInterface</code>	<code>string</code>	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
<code>lun</code>	<code>integer</code>	lun represents iSCSI Target Lun number.

Property	Type	Description
<code>portals</code>	<code>array</code>	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
<code>secretRef</code>	<code>object</code>	secretRef is the CHAP Secret for iSCSI target and initiators authentication
<code>targetPortal</code>	<code>string</code>	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

`.spec.workloads[].template.spec.volumes[].iscsi.portals`

Description

portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

Type

`array`

`.spec.workloads[].template.spec.volumes[].iscsi.portals[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].iscsi.secretRef`

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].nfs`

Description

nfs represents an NFS mount on the host that shares a pod's lifetime More info:
<https://kubernetes.io/docs/concepts/storage/volumes#nfs>

Type

object

Required

`path`

`server`

Property	Type	Description
<code>path</code>	<code>string</code>	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
<code>server</code>	<code>string</code>	<code>server</code> is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

`.spec.workloads[].template.spec.volumes[].persistentVolumeClaim`

Description

`persistentVolumeClaimVolumeSource` represents a reference to a `PersistentVolumeClaim` in the same namespace. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

`object`

Required

`claimName`

Property	Type	Description
<code>claimName</code>	<code>string</code>	<code>claimName</code> is the name of a <code>PersistentVolumeClaim</code> in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> Will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> . Default <code>false</code> .

`.spec.workloads[].template.spec.volumes[].photonPersistentDisk`

Description

`photonPersistentDisk` represents a `PhotonController` persistent disk attached and mounted on kubelets host machine

Type

`object`

Required

`pdID`

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>pdID</code>	<code>string</code>	<code>pdID</code> is the ID that identifies Photon Controller persistent disk

`.spec.workloads[].template.spec.volumes[].portworxVolume`

Description

portworxVolume represents a portworx volume attached and mounted on kubelets host machine

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

.spec.workloads[].template.spec.volumes[].projected

Description

projected items for all in one resources secrets, configmaps, and downward API

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	<code>defaultMode</code> are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like <code>fsGroup</code>, and the result can be other mode bits set.
<code>sources</code>	<code>array</code>	<code>sources</code> is the list of volume projections. Each entry in this list handles one source.

`.spec.workloads[].template.spec.volumes[].projected.sources`

Description

`sources` is the list of volume projections. Each entry in this list handles one source.

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[]`

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

`object`

Property	Type	Description
<code>clusterTrustBundle</code>	<code>object</code>	ClusterTrustBundle allows a pod to access the <code>.spec.trustBundle</code> field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.
<code>configMap</code>	<code>object</code>	configMap information about the configMap data to project
<code>downwardAPI</code>	<code>object</code>	downwardAPI information about the downwardAPI data to project
<code>secret</code>	<code>object</code>	secret information about the secret data to project

Property	Type	Description
<code>serviceAccountToken</code>	<code>object</code>	serviceAccountToken is information about the serviceAccountToken data to project

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle`

Description

ClusterTrustBundle allows a pod to access the `.spec.trustBundle` field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.`

Type

`object`

Required

`path`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Property	Type	Description
<code>name</code>	<code>string</code>	Select a single ClusterTrustBundle by object name. Mutually-exclusive with <code>signerName</code> and <code>labelSelector</code> .
<code>optional</code>	<code>boolean</code>	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using <code>name</code> , then the named ClusterTrustBundle is allowed not to exist. If using <code>signerName</code> , then the combination of <code>signerName</code> and <code>labelSelector</code> is allowed to match zero ClusterTrustBundles.
<code>path</code>	<code>string</code>	Relative path from the volume root to write the bundle.
<code>signerName</code>	<code>string</code>	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with <code>name</code> . The contents of all selected ClusterTrustBundles will be unified and deduplicated.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector`

Description

Select all ClusterTrustBundles that match this label selector. Only has effect if `signerName` is set. Mutually-exclusive with `name`. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[ ]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]`

Type

string

.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.volumes[].projected.sources[].configMap

Description

configMap information about the configMap data to project

Type

object

Property	Type	Description
<code>items</code>	<code>array</code>	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
<code>optional</code>	<code>boolean</code>	optional specify whether the ConfigMap or its keys must be defined

`.spec.workloads[].template.spec.volumes[].projected.sources[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will

error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
<code>path</code>	<code>string</code>	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI`

Description

downwardAPI information about the downwardAPI data to project

Type

`object`

Property	Type	Description
<code>items</code>	<code>array</code>	Items is a list of DownwardAPIVolume file

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items`

Description

Items is a list of DownwardAPIVolume file

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
fieldRef	object	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory,

Property	Type	Description
		requests.cpu and requests.memory) are currently supported.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

object

Required

fieldPath

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	☐	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

.spec.workloads[].template.spec.volumes[].projected.sources[].secret

Description

secret information about the secret data to project

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified

Property	Type	Description
		paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	optional field specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].projected.sources[].serviceAccountToken`

Description

serviceAccountToken is information about the serviceAccountToken data to project

Type

object

Required

path

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours.Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

.spec.workloads[].template.spec.volumes[].quobyte

Description

quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Type

object

Required

registry

volume

Property	Type	Description
group	string	group to map volume access to Default is no group
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user
volume	string	volume is a string that references an already created Quobyte volume by name.

.spec.workloads[].template.spec.volumes[].rbd

Description

rbd represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/rbd/README.md>

Type

object

Required

image

monitors

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Property	Type	Description
monitors	array	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
user	string	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

.spec.workloads[].template.spec.volumes[].rbd.monitors

Description

monitors is a collection of Ceph monitors. More info:

<https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

array

`.spec.workloads[].template.spec.volumes[].rbd.monitors[]`

Type

string

`.spec.workloads[].template.spec.volumes[].rbd.secretRef`

Description

secretRef is name of the authentication secret for RBDUser. If provided overrides keyring.

Default is nil. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].scaleIO`

Description

scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Type

`object`

Required

`gateway``secretRef``system`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
<code>gateway</code>	<code>string</code>	gateway is the host address of the ScaleIO API Gateway.
<code>protectionDomain</code>	<code>string</code>	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
<code>readOnly</code>	<code>boolean</code>	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>secretRef</code>	<code>object</code>	secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.
<code>sslEnabled</code>	<code>boolean</code>	sslEnabled Flag enable/disable SSL communication with Gateway, default false

Property	Type	Description
<code>storageMode</code>	<code>string</code>	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
<code>storagePool</code>	<code>string</code>	storagePool is the ScaleIO Storage Pool associated with the protection domain.
<code>system</code>	<code>string</code>	system is the name of the storage system as configured in ScaleIO.
<code>volumeName</code>	<code>string</code>	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

`.spec.workloads[].template.spec.volumes[].scaleIO.secretRef`

Description

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty.

Property	Type	Description
		Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].secret`

Description

secret represents a secret that should populate this volume. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#secret>

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume

Property	Type	Description
		setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>optional</code>	<code>boolean</code>	optional field specify whether the Secret or its keys must be defined
<code>secretName</code>	<code>string</code>	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

`.spec.workloads[].template.spec.volumes[].secret.items`

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

`array`

`.spec.workloads[].template.spec.volumes[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

`object`

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.workloads[].template.spec.volumes[].storageos

Description

storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.

Type

object

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
<code>volumeNamespace</code>	<code>string</code>	<code>volumeNamespace</code> specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set <code>VolumeName</code> to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

`.spec.workloads[].template.spec.volumes[].storageos.secretRef`

Description

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].vsphereVolume`

Description

vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Type

object

Required

`volumePath`

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>storagePolicyID</code>	<code>string</code>	<code>storagePolicyID</code> is the storage Policy Based Management (SPBM) profile ID associated with the <code>StoragePolicyName</code> .
<code>storagePolicyName</code>	<code>string</code>	<code>storagePolicyName</code> is the storage Policy Based Management (SPBM) profile name.
<code>volumePath</code>	<code>string</code>	<code>volumePath</code> is the path that identifies vSphere volume vmrk

.status

Description

`ComponentStatus` defines the observed state of Connector

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	<code>Annotations</code> is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is

Property	Type	Description
		roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See:
<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
status	string	Status of the condition, one of True, False, Unknown.

Property	Type	Description
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s`
 - `DELETE` : delete collection of ConnectorsK8S
 - `GET` : list objects of kind ConnectorsK8S
 - `POST` : create a new ConnectorsK8S
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s/{name}`
 - `DELETE` : delete the specified ConnectorsK8S
 - `GET` : read the specified ConnectorsK8S
 - `PATCH` : partially update the specified ConnectorsK8S
 - `PUT` : replace the specified ConnectorsK8S
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s/{name}/status`
 - `GET` : read status of the specified ConnectorsK8S
 - `PATCH` : partially update status of the specified ConnectorsK8S
 - `PUT` : replace status of the specified ConnectorsK8S

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s`

HTTP method

DELETE

Description

delete collection of ConnectorsK8S

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ConnectorsK8S

HTTP responses

HTTP code	Response body
200 - OK	ConnectorsK8SList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ConnectorsK8S

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsK8S</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
201 - Created	<code>ConnectorsK8S</code> schema
202 - Accepted	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s/{name}

HTTP method

DELETE

Description

delete the specified ConnectorsK8S

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ConnectorsK8S

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ConnectorsK8S

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ConnectorsK8S

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsK8S</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
201 - Created	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s/{name}/status`

HTTP method

GET

Description

read status of the specified ConnectorsK8S

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ConnectorsK8S

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectorsK8S

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsK8S</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
201 - Created	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

ConnectorsOCI

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsOCI is the Schema for the ConnectorsOCIs API.

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsOCISpec defines the desired state of ConnectorsOCI.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsOCISpec defines the desired state of ConnectorsOCI.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>expose</code>	<code>object</code>	Expose defines the expose configuration for the connector
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

.spec.annotations

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

.spec.expose

Description

Expose defines the expose configuration for the connector

Type

object

Required

type

Property	Type	Description
domain	string	Domain specifies the access domain for the connector service. Required when Type is Ingress or NodePort: • For Ingress: Set to your ingress domain (e.g., "connectors.example.com") • For NodePort: Set to your cluster node IP (e.g., "192.168.1.100")
ingress	object	Ingress configuration for external access via ingress controller. Required when Type is Ingress.
nodePort	object	NodePort configuration for external access via node port. Required when Type is NodePort.
type	string	Type specifies the exposure method for the OCI connector service. • ClusterIP: Internal cluster access only (default) • NodePort: Access via node IP and port from outside the cluster • Ingress: Access via ingress controller with domain name

.spec.expose.ingress

Description

Ingress configuration for external access via ingress controller. Required when Type is Ingress.

Type

object

Property	Type	Description
<code>ingressClassName</code>	<code>string</code>	IngressClassName specifies the ingress controller class to use. Leave empty to use the default ingress controller. eg: "nginx"
<code>secretName</code>	<code>string</code>	SecretName specifies the TLS secret for HTTPS access. Example: "connectors-tls-secret"

.spec.expose.nodePort

Description

NodePort configuration for external access via node port. Required when Type is NodePort.

Type

object

Property	Type	Description
<code>port</code>	<code>integer</code>	Port specifies the node port number for external access. Must be between 30000-32767 (Kubernetes node port range). Recommended: Use ports 30000-31000 to avoid conflicts. Example: 30080 for HTTP, 30443 for HTTPS

.spec.labels

Description

Labels overrides labels for the namespace and its template.

Type

object

.spec.workloads

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.

Property	Type	Description
<code>template</code>	<code>object</code>	Template describes the pods that will be created. The only allowed <code>template.spec.restartPolicy</code> value is "Always".

`.spec.workloads[].template`

Description

Template describes the pods that will be created. The only allowed `template.spec.restartPolicy` value is "Always".

Type

`object`

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
<code>spec</code>	<code>object</code>	Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status ↗

`.spec.workloads[].template.spec`

Description

Specification of the desired behavior of the pod. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

Type

object

Required

containers

Property	Type	Description
activeDeadlineSeconds	integer	Optional duration in seconds the pod may StartTime before the system will actively try to mark it failed and kill any containers. Value must be a positive integer.
affinity	object	If specified, the pod's scheduling constraints.
automountServiceAccountToken	boolean	AutomountServiceAccountToken indicates whether an automounted service account token should be automatically mounted.
containers	array	List of containers belonging to the pod. Containers can never be removed. There must be at least one container.
dnsConfig	object	Specifies the DNS parameters of a pod. Parameters specified in the pod spec overrides the generated DNS configuration based on the hostNetwork and hostIPC fields.
dnsPolicy	string	Set DNS policy for the pod. Defaults to "ClusterFirst". Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default', or 'None'. Parameters specified in the pod spec overrides the default value. Parameters specified in the DNSConfig will be merged with the policy parameters. The 'Default' policy means that the pod's DNS options are merged with the default/pod-level DNS options set along with hostNetwork, y to 'ClusterFirstWithHostNet'.

Property	Type	Description
<code>enableServiceLinks</code>	<code>boolean</code>	<code>EnableServiceLinks</code> indicates whether info injected into pod's environment variables, i Optional: Defaults to true.
<code>ephemeralContainers</code>	<code>array</code>	List of ephemeral containers run in this pod an existing pod to perform user-initiated ac cannot be specified when creating a pod, a the pod spec. In order to add an ephemera pod's <code>ephemeralcontainers</code> subresource.
<code>hostAliases</code>	<code>array</code>	<code>HostAliases</code> is an optional list of hosts and hosts file if specified.
<code>hostIPC</code>	<code>boolean</code>	Use the host's ipc namespace. Optional: D
<code>hostNetwork</code>	<code>boolean</code>	Host networking requested for this pod. Us option is set, the ports that will be used mu
<code>hostPID</code>	<code>boolean</code>	Use the host's pid namespace. Optional: D
<code>hostUsers</code>	<code>boolean</code>	Use the host's user namespace. Optional: present, the pod will be run in the host use needs a feature only available to the host u kernel module with <code>CAP_SYS_MODULE</code> . \ created for the pod. Setting false is useful i vulnerabilities even allowing users to run th

Property	Type	Description
		having root privileges on the host. This field is not applicable to servers that enable the UserNamespacesS
hostname	string	Specifies the hostname of the Pod. If not specified, the pod's hostname will be set to a system-defined value.
imagePullSecrets	array	ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images listed in the pod's containers. These secrets will be passed to individual pods and used for pulling the images. More info: https://kubernetes.io/docs/concepts/configuration/secret imagepullsecrets-on-a-pod ↗
initContainers	array	List of initialization containers belonging to the pod. Init containers are designed to allow users to run containerized "scripts" during pod start-up. This can be used to build the app directory for a particular application, to update or replace a library in a container, to install additional permissions on a pod, etc. Pod init containers must have Lifecycle actions, Startup probes, and Liveness probes (other probes must not be set). The resourceRequirements field is optional for init containers. Limits are applied to init containers in a pod. The initContainers field cannot be added or removed. Cannot be used on a pod that has initContainers, but that was created before https://kubernetes.io/docs/concepts/workloads/pods/init-containers/.

Property	Type	Description
nodeName	string	nodeName indicates in which node this pod candidate for scheduling by the scheduler field is set, the kubelet for this node becomes pod. This field should not be used to express on a specific node. https://kubernetes.io/docs/concepts/scheduling/kubernetes-scheduler.html#node-name
nodeSelector	object	NodeSelector is a selector which must be true for the pod to be scheduled on a node. NodeSelector and nodeSelectorExact are mutually exclusive. NodeSelector is a selector which must match a node's labels for the pod to be scheduled on it. info: https://kubernetes.io/docs/concepts/scheduling/kubernetes-scheduler.html#node-selector
os	object	Specifies the OS of the containers in the pod. If this is set, the pod is restricted if this is set. If the OS field is set to linux, the following fields are required: securityContext.windowsOptions If the OS field is set to windows, following fields are required: - spec.hostPID - spec.hostIPC - spec.hostUsers - spec.securityContext.appArmorProfile - spec.securityContext.seLinuxOptions - spec.securityContext.seccompProfile - spec.securityContext.fsGroup - spec.securityContext.fsGroupChangePolicy - spec.securityContext.sysctls - spec.shareProcessNamespace

Property	Type	Description
		- spec.securityContext.runAsUser - spec.securityContext.runAsGroup - spec.securityContext.supplementalGroups - spec.securityContext.supplementalGroups - spec.containers[*].securityContext.appArmorProfile - spec.containers[*].securityContext.seLinuxOptions - spec.containers[*].securityContext.seccompProfile - spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readOnlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsNonRoot - spec.containers[*].securityContext.runAsNonRoot
overhead	object	Overhead represents the resource overhead of a Pod given RuntimeClass. This field will be automatically set by RuntimeClass admission controller. If the FeatureGate is enabled, overhead must not be set in Pod admission controller will reject Pod creation if overhead is already set. If RuntimeClass is configured with overhead, it will be set to the value defined in the corresponding RuntimeClass. If not set, overhead will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/1443-pod-overhead
preemptionPolicy	string	PreemptionPolicy is the Policy for preempting lower priority Pods. It can be set to PreemptLowerPriority or Never. Defaults to PreemptLowerPriority.

Property	Type	Description
<code>priority</code>	<code>integer</code>	The priority value. Various system compon the pod. When Priority Admission Controlle setting this field. The admission controller j PriorityClassName. The higher the value, t
<code>priorityClassName</code>	<code>string</code>	If specified, indicates the pod's priority. "sy: critical" are two special keywords which inc former being the highest priority. Any other PriorityClass object with that name. If not s or zero if there is no default.
<code>readinessGates</code>	<code>array</code>	If specified, all readiness gates will be eval when all its containers are ready AND all c gates have status equal to "True" More info https://git.k8s.io/enhancements/keps/sig-nr
<code>resourceClaims</code>	<code>array</code>	ResourceClaims defines which ResourceC before the Pod is allowed to start. The resc containers which consume them by name. This is an alpha field and requires enabling feature gate. This field is immutable.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy for all containers within the pod. Some contexts have some contexts, only a subset of those values are allowed. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle/#restart-policy
<code>runtimeClassName</code>	<code>string</code>	RuntimeClassName refers to a RuntimeClass object in the kubelet registry which should be used to run this pod. If no class is named, the pod will not be run. If unspecified, the default runtime handler will be used, which is an implicit class with the name "default". More info: https://kubernetes.io/docs/concepts/workloads/pods/runtime-handlers/
<code>schedulerName</code>	<code>string</code>	If specified, the pod will be dispatched by specified scheduler. Otherwise, the pod will be dispatched by default scheduler.
<code>schedulingGates</code>	<code>array</code>	SchedulingGates is an opaque list of values that must all be satisfied in order for the pod to be scheduled. If schedulingGates is not empty, the pod will not be scheduled and the scheduler will not attempt to reschedule the pod. SchedulingGates can only be set at pod creation and cannot be updated afterwards.
<code>securityContext</code>	<code>object</code>	SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See the documentation for the fields of SecurityContext for their defaults.

Property	Type	Description
<code>serviceAccount</code>	<code>string</code>	DeprecatedServiceAccount is a deprecated property. Deprecated: Use serviceAccountName instead.
<code>serviceAccountName</code>	<code>string</code>	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/service-account/
<code>setHostnameAsFQDN</code>	<code>boolean</code>	If true the pod's hostname will be configured to the pod's leaf name (the default). In Linux containers this means setting the pod's hostname field of the kernel (the nodename) to the pod's hostname. In Windows containers, this means setting the registry value of the pod's hostname to the pod's fully qualified domain name (FQDN). If a pod does not have FQDN, this will be false.
<code>shareProcessNamespace</code>	<code>boolean</code>	Share a single process namespace between all containers in this pod. When this is set containers will be able to view ar and signal each other in the same process namespace. Optional: Default to false.
<code>subdomain</code>	<code>string</code>	If specified, the fully qualified Pod hostname will be <code><hostname>.<subdomain>.<pod namespace>.<domain name></code> . If not specified, the pod will not have a domainname at all.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no graceful termination is required). If the value is nil, the default grace period will be used instead.

Property	Type	Description
		duration in seconds after the processes run signal and the time when the processes are this value longer than the expected cleanup seconds.
<code>tolerations</code>	<code>array</code>	If specified, the pod's tolerations.
<code>topologySpreadConstraints</code>	<code>array</code>	TopologySpreadConstraints describes how pods are spread across topology domains. Scheduler will schedule pods according to the constraints. All topologySpreadConstraints must be satisfied.
<code>volumes</code>	<code>array</code>	List of volumes that can be mounted by containers in this pod. info: https://kubernetes.io/docs/concepts/storage/persistent-storage/

`.spec.workloads[].template.spec.affinity`

Description

If specified, the pod's scheduling constraints

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	<code>object</code>	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Property	Type	Description
		pod(s)).
<code>podAntiAffinity</code>	<code>object</code>	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

`.spec.workloads[].template.spec.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSched

Property	Type	Description
		affinity expressions, € compute a sum by iterating through the elements of this field adding "weight" to the sum if the node matc the corresponding matchExpressions; th node(s) with the high sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirem specified by this field not met at scheduling time, the pod will not scheduled onto the n If the affinity requirem specified by this field cease to be met at sc point during pod execution (e.g. due to update), the system r or may not try to eventually evict the p from its node.

**`.spec.workloads[].template.spec.affinity.nodeAffinity.prefe
redDuringSchedulingIgnoredDuringExecution`**

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is

Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

`object`

Required

`nodeSelectorTerms`

Property	Type	Description
<code>nodeSelectorTerms</code>	<code>array</code>	Required. A list of node selector terms. The terms are ORed.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]`

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector`

orTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity`

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of

Property	Type	Description
		scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, node affinity expressions, etc.). If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm
requiredDuringSchedulingIgnoredDuringExecution	array	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm

Property	Type	Description
		intersected, i.e. all terms must be satisfied.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with

Property	Type	Description
		<code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code>. Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	<code>MismatchLabelKeys</code> is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null

Property	Type	Description
		or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

`yTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

yTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code>. Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector` or

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchL

abelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace`

`eSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

`.spec.workloads[].template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expression specified by this field, it may choose a node that violates one or more the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution anti-affinity expression etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has p

Property	Type	Description
		which matches the corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at the time of scheduling, the pod will not be scheduled on the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm intersected, i.e. all terms must be satisfied.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

Required. A pod affinity term, associated with the corresponding weight.

object

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

finityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values
[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

finityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismat

chLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

paceSelector.matchExpressions[].values[]

Type

string

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paceSelector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces**

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

**.spec.workloads[].template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces[]**

Type

string

.spec.workloads[].template.spec.containers

Description

List of containers belonging to the pod. Containers cannot currently be added or removed.
There must be at least one container in a Pod. Cannot be updated.

Type

array

.spec.workloads[].template.spec.containers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image if this is not provided. Variable references <code>\$(VAR)</code> expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must match the container's environment.

Property	Type	Description
		C_IDENTIFIER. All invalid keys will be reported when the container is starting. When a key exists in multiple sources, the value associated with the last source takes precedence. Values defined by an Env with a duplicate key take precedence. Cannot be updated.
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images field is optional to allow higher level config management to default or override container images in workload Deployments and StatefulSets.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images
lifecycle	object	Actions that the management system should take in response to container lifecycle events. Cannot be updated.
livenessProbe	object	Periodic probe of container liveness. Container v1.4.0 adds the ability to perform an httpGet to the probe endpoint. If the probe fails, the container is restarted. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL).

Property	Type	Description
		Cannot be updated.
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address on the container will be accessible from the network. Must be an array with max length 70. Strategic merge patch may corrupt the information. See https://github.com/kubernetes/kubernetes/issues/20380 for more info. Cannot be updated.
readinessProbe	object	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
resizePolicy	array	Resources resize policy for the container.
resources	object	Compute Resources required by this container. Cannot be updated. More info: https://kubernetes.io/docs/concepts/configuration/assigning-resources-to-containers/
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and in that case the only allowed value is "Always". For non-init containers the default value is "Always" if the field is not specified, the restart behavior is "OnFailure" if set to "OnFailure", and "Never" otherwise.

Property	Type	Description
		Pod's restart policy and the container type. Setting RestartPolicy as "Always" for the init container will have the following effect: this init container will be continuously restarted until all regular containers have terminated. Once all regular containers have completed, all init containers with RestartPolicy "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar". Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next container starts immediately after this init container completes, after any startupProbe has successfully completed.
<code>securityContext</code>	<code>object</code>	SecurityContext defines the security options the container will be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/
<code>startupProbe</code>	<code>object</code>	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully. If this probe fails, the Pod will be restarted. If the livenessProbe fails. This can be used to provide probe parameters at the beginning of a Pod's life cycle. It might take a long time to load data or warm a cache before steady-state operation. This cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin data at container runtime. If this is not set, reads from stdin will return EOF.

Property	Type	Description
		container will always result in EOF. Default is false
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When false, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process from stdin will never receive an EOF. Default is false
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container termination message will be written is mounted in the container's filesystem. Message written is intended to be of final status, such as an assertion failure message. Message length across all containers will be limited by the node if greater than 4096 bytes. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. AllLogs will use the contents of terminationMessagePath on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 2 lines, whichever is smaller. Defaults to File. Cannot be updated.

Property	Type	Description
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be u container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystems updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be co container image. Cannot be updated.

`.spec.workloads[].template.spec.containers[].args`

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$ (VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

`array`

`.spec.workloads[].template.spec.containers[].args[]`

Type

string

`.spec.workloads[].template.spec.containers[].command`

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$ (VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.containers[].command[]`

Type

string

`.spec.workloads[].template.spec.containers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	Source for the environment variable's value. Cannot be used if value is not empty.

.spec.workloads[].template.spec.containers[].env[].valueFrom

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

object

Property	Type	Description
<code>configMapKeyRef</code>	object	Selects a key of a ConfigMap.
<code>fieldRef</code>	object	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
<code>resourceFieldRef</code>	object	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	object	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.containers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

object

Required

key

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap or its key must be defined

.spec.workloads[].template.spec.containers[].env[].valueFrom.fieldRef

Description

Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Type

object

Required

fieldPath

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.containers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>string</code>	Specifies the output format of the exposed resources, defaults to "1"

Property	Type	Description
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.containers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key of the secret to select from. Must be a valid secret key.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.containers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.containers[].envFrom[]`

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	<code>object</code>	The ConfigMap to select from
<code>prefix</code>	<code>string</code>	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	<code>object</code>	The Secret to select from

`.spec.workloads[].template.spec.containers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap must be defined

.spec.workloads[].template.spec.containers[].envFrom[].secretRef

Description

The Secret to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.containers[].lifecycle`

Description

Actions that the management system should take in response to container lifecycle events.
Cannot be updated.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
<code>preStop</code>	<code>object</code>	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the

Property	Type	Description
		container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗

`.spec.workloads[].template.spec.containers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>string</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name``value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.containers[].lifecycle.postStart.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perform.
<code>sleep</code>	<code>object</code>	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	<code>object</code>	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.containers[].lifecycle.preStop.tcpSocket`

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].livenessProbe`

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies an action involving a GRP
<code>httpGet</code>	object	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	integer	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes
<code>periodSeconds</code>	integer	How often (in seconds) to perform the prot 10 seconds. Minimum value is 1.
<code>successThreshold</code>	integer	Minimum consecutive successes for the pr considered successful after having failed. I

Property	Type	Description
		Must be 1 for liveness and startup. Minimum
tcpSocket	object	TCPSocket specifies an action involving a
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod need gracefully upon probe failure. The grace period duration in seconds after the processes running in the pod are sent a termination signal and the time until the processes are forcibly halted with a kill signal. It specifies that the container managed liveness probes (kube-probes) and terminationGracePeriodSeconds will be used to terminate the pod. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].livenessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].livenessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].ports`

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.containers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.containers[].readinessProbe`

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period in seconds after the processes running on the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/probe-lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.containers[].readinessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.containers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.containers[].readinessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.workloads[].template.spec.containers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.containers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.containers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>	<code>int32</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.containers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.containers[].resources`

Description

Compute Resources required by this container. Cannot be updated. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.containers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.containers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.containers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.containers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.containers[].securityContext`

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	object	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.containers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.containers[].securityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.containers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.containers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.containers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.containers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.containers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes

`.spec.workloads[].template.spec.containers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.containers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.containers[].startupProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.containers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.containers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.containers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.containers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.containers[].startupProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.containers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.containers[].volumeDevices[]`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.containers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.containers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain '..'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	string	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPathExpr	string	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

.spec.workloads[].template.spec.dnsConfig

Description

Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy.

Type

object

Property	Type	Description
<code>nameservers</code>	<code>array</code>	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
<code>options</code>	<code>array</code>	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.
<code>searches</code>	<code>array</code>	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

`.spec.workloads[].template.spec.dnsConfig.nameservers`

Description

A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.

Type

`array`

`.spec.workloads[].template.spec.dnsConfig.nameservers[]`

Type

`string`

`.spec.workloads[].template.spec.dnsConfig.options`

Description

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type

array

`.spec.workloads[].template.spec.dnsConfig.options[]`

Description

PodDNSConfigOption defines DNS resolver options of a pod.

Type

object

Property	Type	Description
name	string	Required.
value	string	

`.spec.workloads[].template.spec.dnsConfig.searches`

Description

A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

Type

array

`.spec.workloads[].template.spec.dnsConfig.searches[]`

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers`

Description

List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[]`

Description

An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.

Type

`object`

Required

`name`

Property	Type	Description
<code>args</code>	<code>array</code>	Arguments to the entrypoint. The image's CMD is not provided. Variable references \$(VAR_NAME) using the container's environment. If a variable c

Property	Type	Description
		resolved, the reference in the input string will be Double \$\$ are reduced to a single \$, which allow the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" string literal "\$(VAR_NAME)". Escaped reference expanded, regardless of whether the variable ex Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
<code>command</code>	<code>array</code>	Entrypoint array. Not executed within a shell. The ENTRYPOINT is used if this is not provided. Vari \$(VAR_NAME) are expanded using the containe If a variable cannot be resolved, the reference in will be unchanged. Double \$\$ are reduced to a s allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ Escaped references will never be expanded, reg whether the variable exists or not. Cannot be up https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
<code>env</code>	<code>array</code>	List of environment variables to set in the contain updated.

Property	Type	Description
<code>envFrom</code>	<code>array</code>	List of sources to populate environment variables for the container. The keys defined within a source must match the C_IDENTIFIER. All invalid keys will be reported as an error when the container is starting. When a key exists across multiple sources, the value associated with the last source in the list takes precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
<code>image</code>	<code>string</code>	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images
<code>imagePullPolicy</code>	<code>string</code>	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images
<code>lifecycle</code>	<code>object</code>	Lifecycle is not allowed for ephemeral containers
<code>livenessProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>name</code>	<code>string</code>	Name of the ephemeral container specified as a pod's spec. This name must be unique among all containers, and ephemeral containers.

Property	Type	Description
<code>ports</code>	<code>array</code>	Ports are not allowed for ephemeral containers.
<code>readinessProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	Resources are not allowed for ephemeral containers use spare resources already allocated
<code>restartPolicy</code>	<code>string</code>	Restart policy for the container to manage the restart of each container within a pod. This may only be set on non-ephemeral containers. You cannot set this field on ephemeral containers.
<code>securityContext</code>	<code>object</code>	Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the SecurityContext override the equivalent fields of PodSecurityContext.
<code>startupProbe</code>	<code>object</code>	Probes are not allowed for ephemeral containers
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for stdin in container runtime. If this is not set, reads from stdin in container will always result in EOF. Default is false

Property	Type	Description
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When false, the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on pod start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the pod is restarted. If this flag is false, a container process starting from stdin will never receive an EOF. Default is false
<code>targetContainerName</code>	<code>string</code>	If set, the name of the container from PodSpec to which the ephemeral container targets. The ephemeral container will then run in the namespaces (IPC, PID, etc) of this container. If not set, then the ephemeral container uses the namespaces of the Pod spec. The container runtime must implement support for namespace targeting. If the runtime does not support namespace targeting, the result of setting this field is undefined.
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container's termination message will be written is mounted in the container's filesystem. Message written is intended to be consumed by Kubernetes. Max length of the message across all containers will be limited by the node. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. A container can indicate its termination message will be used to its exit code by specifying a policy of 'FallbackToLogsOnError'.

Property	Type	Description
		container status message on both success and failure. FallbackToLogsOnError will use the last chunk of output if the termination message file is empty and exited with an error. The log output is limited to 2048 lines, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for stdin. Requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used within the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be configurable by the container image. Cannot be updated.

`.spec.workloads[].template.spec.ephemeralContainers[].args`

Description

Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e.

"\$\$ (VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].args[]

Type

string

.spec.workloads[].template.spec.ephemeralContainers[].command

Description

Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].command[]

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].env[].`

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. Must be a C_IDENTIFIER.
<code>value</code>	<code>string</code>	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME)

Property	Type	Description
		syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

object

Required

resource

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>		Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[]`

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	The ConfigMap to select from
prefix	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
secretRef	object	The Secret to select from

.spec.workloads[].template.spec.ephemeralContainers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the ConfigMap must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].envFrom[].secretRef`

Description

The Secret to select from

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle`

Description

Lifecycle is not allowed for ephemeral containers.

Type

object

Property	Type	Description
<code>postStart</code>	object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks
<code>preStop</code>	object	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].ifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].ifecycle.postStart.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be

Property	Type	Description
		an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility.

Property	Type	Description
		There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.exec.command[]

Type

string

.spec.workloads[].template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.

Property	Type	Description
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

- `name`
- `value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.ephemeralContainers[].lifecyle.preStop.tcpSocket`

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when

tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimum
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs gracefully upon probe failure. The grace period duration in seconds after the processes run

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u:
timeoutSeconds	integer	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. <a href="https://kubernetes.io/docs/concepts/worklo
lifecycle#container-probes">https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`

Required

`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.ephemeralContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].ports`

Description

Ports are not allowed for ephemeral containers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe`

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. Defaults to 30 seconds. spec.terminationGracePeriodSeconds is used when specified.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/probe-lifecycle#container-probes

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.ephemeralContainers[].readinessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].resizePolicy[]`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName`

`restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.ephemeralContainers[].resources`

Description

Resources are not allowed for ephemeral containers. Ephemeral containers use spare resources already allocated to the pod.

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.ephemeralContainers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.ephemeralContainers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.ephemeralContainers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext`

Description

Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.ephemeralContainers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe`

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.
failureThreshold	integer	Minimum consecutive failures for the probe considered failed after having succeeded.

Property	Type	Description
		Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRP
httpGet	object	HTTPGet specifies the http request to perf
initialDelaySeconds	integer	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
periodSeconds	integer	How often (in seconds) to perform the prot 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
tcpSocket	object	TCPSocket specifies an action involving a
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim

Property	Type	Description
		process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via kill (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to specify the shell in the container's image.

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>string</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.ephemeralContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.ephemeralContainers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].volumeDevices[]`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.ephemeralContainers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.ephemeralContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.hostAliases`

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

`array`

`.spec.workloads[].template.spec.hostAliases[]`

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type

object

Required

ip

Property	Type	Description
hostnames	array	Hostnames for the above IP address.
ip	string	IP address of the host file entry.

.spec.workloads[].template.spec.hostAliases[].hostnames**Description**

Hostnames for the above IP address.

Type

array

.spec.workloads[].template.spec.hostAliases[].hostnames
[]**Type**

string

.spec.workloads[].template.spec.imagePullSecrets**Description**

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info:

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

Type

array

`.spec.workloads[].template.spec.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.initContainers`

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init

containers cannot currently be added or removed. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

Type

array

```
.spec.workloads[].template.spec.initContainers[]
```

Description

A single application container that you want to run within a pod.

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be resolved, the reference in the input string will be replaced by the resolved value. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" results in a string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be replaced by the resolved value.

Property	Type	Description
		the input string will be unchanged. Double \$\$ are single \$, which allows for escaping the \$(VAR_NAME) i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded to their values regardless of whether the variable exists or not. If the variable is not set or updated, the expansion will result in an empty string and the variable will be treated as updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container-command-in-a-shell/
env	array	List of environment variables to set in the container. If specified, the env file must be an array of key=value pairs. The pairs defined within the array must be valid environment variables, ie. identifiers which can be used to set environment variables in the container. All invalid keys will be reported as an error when the container is starting. When a key exists in multiple sources, the value associated with the last source with the same precedence will be the final value. Values defined by an Env with a higher precedence take precedence. Cannot be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must match those of the container C_IDENTIFIER. All invalid keys will be reported as an error when the container is starting. When a key exists in multiple sources, the value associated with the last source with the same precedence will be the final value. Values defined by an Env with a higher precedence take precedence. Cannot be updated.
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images/ field is optional to allow higher level config management to default or override container images in workload Deployments and StatefulSets.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images/#image-pull-policy

Property	Type	Description
		https://kubernetes.io/docs/concepts/containers/images/ ↗
lifecycle	object	Actions that the management system should take on container lifecycle events. Cannot be updated.
livenessProbe	object	Periodic probe of container liveness. Container v1 the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
name	string	Name of the container specified as a DNS_LABEL. Container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. You must specify a port here to allow the container to be accessible from the network. Most containers have a single port in which they listen on the default "0.0.0.0" address. Mutating the ports array with strategic merge patch may corrupt the data. See https://github.com/kubernetes/kubernetes/issues/108132 for more information. Cannot be updated.
readinessProbe	object	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗

Property	Type	Description
		https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	Compute Resources required by this container. (More info: https://kubernetes.io/docs/concepts/configuration/resource-limits)
<code>restartPolicy</code>	<code>string</code>	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers and only allowed value is "Always". For non-init containers, if this field is not specified, the restart policy is default to the Pod's restart policy and the container type. Setting RestartPolicy to "Always" for the init container will have the following effect: this init container will be continually restarted until all regular containers have terminated. Once all regular containers have completed, all init containers with "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar". Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container after any startupProbe has successfully completed.
<code>securityContext</code>	<code>object</code>	SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/concepts/security/pod-security-context/

Property	Type	Description
		https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ ↗
startupProbe	object	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully. If this probe fails, the Pod will be restarted. This can be used to probe parameters at the beginning of a Pod's life cycle. This cannot be updated. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
stdin	boolean	Whether this container should allocate a buffer for stdin data in its container runtime. If this is not set, reads from stdin within the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When set to true, the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is only open until the first client attaches to stdin, which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process that opens stdin from stdout will never receive an EOF. Default is false.
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted in the container's filesystem. Message written is intended to be consumed by a downstream component. Default path is /dev/termination-log, which requires mount access for the container and is only available on Linux.

Property	Type	Description
		final status, such as an assertion failure message truncated by the node if greater than 4096 bytes. message length across all containers will be limited. Defaults to /dev/termination-log. Cannot be updated.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be handled. Will use the contents of terminationMessagePath for container status message on both success and failure. FallbackToLogsOnError will use the last chunk of output if the termination message file is empty and exited with an error. The log output is limited to 2 lines, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for stdin. Requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used within the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Updated after the pod is updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the runtime's default will be used, which might be configurable for the container image. Cannot be updated.

`.spec.workloads[].template.spec.initContainers[].args`

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$ (VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.initContainers[].args[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].command`

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double `$$` are reduced to a single `$`, which allows for escaping the `$(VAR_NAME)` syntax: i.e. `"$$ (VAR_NAME)"` will produce the string literal `"$(VAR_NAME)"`. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

`.spec.workloads[].template.spec.initContainers[].command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].env`

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.initContainers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. Must be a C_IDENTIFIER.
<code>value</code>	<code>string</code>	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input

Property	Type	Description
		string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

object

Required

resource

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>int</code>	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.initContainers[].env[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.initContainers[].envFrom`

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

`.spec.workloads[].template.spec.initContainers[].envFrom`



Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
<code>configMapRef</code>	object	The ConfigMap to select from
<code>prefix</code>	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	object	The Secret to select from

`.spec.workloads[].template.spec.initContainers[].envFrom[].configMapRef`

Description

The ConfigMap to select from

Type

object

Property	Type	Description
<code>name</code>	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap must be defined

`.spec.workloads[].template.spec.initContainers[].envFrom[].secretRef`

Description

The Secret to select from

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.workloads[].template.spec.initContainers[].lifecycle`

Description

Actions that the management system should take in response to container lifecycle events.

Cannot be updated.

Type

object

Property	Type	Description
postStart	object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks
preStop	object	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart`

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the

container blocks until the hook completes. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.
httpGet	object	HTTPGet specifies the http request to perform.
sleep	object	Sleep represents the duration that the container should sleep before being terminated.
tcpSocket	object	Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

**.spec.workloads[].template.spec.initContainers[].lifecycle.
postStart.httpGet.httpHeaders**

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.workloads[].template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info:

<https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks>

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>httpGet</code>	object	HTTPGet specifies the http request to perform.
<code>sleep</code>	object	Sleep represents the duration that the container should sleep before being terminated.
<code>tcpSocket</code>	object	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when tcp handler is specified.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec`

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.exec.command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.

Property	Type	Description
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.sleep`

Description

Sleep represents the duration that the container should sleep before being terminated.

Type

`object`

Required

`seconds`

Property	Type	Description
<code>seconds</code>	<code>integer</code>	Seconds is the number of seconds to sleep.

`.spec.workloads[].template.spec.initContainers[].lifecycle.preStop.tcpSocket`

Description

Deprecated. TCP Socket is NOT supported as a LifecycleHandler and kept for the backward compatibility. There are no validation of this field and lifecycle hooks will fail in runtime when

tcp handler is specified.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.workloads[].template.spec.initContainers[].livenessProbe

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

object

Property	Type	Description
exec	object	Exec specifies the action to take.

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes ↗
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sigi value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u:
timeoutSeconds	integer	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. <a href="https://kubernetes.io/docs/concepts/worklo
lifecycle#container-probes">https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.workloads[].template.spec.initContainers[].livenessProbe.exec

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.workloads[].template.spec.initContainers[].livenessProbe.exec.command[]`

Type

string

`.spec.workloads[].template.spec.initContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`**Required**`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type`object`**Required**`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].ports`

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.workloads[].template.spec.initContainers[].readinessProbe`

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via <code>TERM</code> (no opportunity to shut down). This is a beta feature and requires enabling <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. Defaults to 30 seconds. <code>spec.terminationGracePeriodSeconds</code> is used if not set.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/probe-lifecycle#container-probes

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

`.spec.workloads[].template.spec.initContainers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].resizePolicy`

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.

Property	Type	Description
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

`.spec.workloads[].template.spec.initContainers[].resource` **S**

Description

Compute Resources required by this container. Cannot be updated. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

Property	Type	Description
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.initContainers[].resourceClaims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].resourceClaims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.workloads[].template.spec.initContainers[].resource` `s.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.initContainers[].resource` `s.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.workloads[].template.spec.initContainers[].securityContext`

Context

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Type

`object`

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of

Property	Type	Description
		capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.initContainers[].securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

object

Property	Type	Description
add	array	Added capabilities

Property	Type	Description
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.add`

Description

Added capabilities

Type

`array`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

`string`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

`array`

`.spec.workloads[].template.spec.initContainers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.workloads[].template.spec.initContainers[].securityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.initContainers[].securityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container

Property	Type	Description
		runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.initContainers[].securityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same

Property	Type	Description
		effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.initContainers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes>

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRP
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perf
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/workload-life-cycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe considered successful after having failed. Must be 1 for liveness and startup. Minimu
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies an action involving a

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod need gracefully upon probe failure. The grace period duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It must be longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via SIGKILL (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].startupProbe.exec.command[]`

Type

`string`

`.spec.workloads[].template.spec.initContainers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.workloads[].template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>int</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.workloads[].template.spec.initContainers[].volumeDevices`

Description

`volumeDevices` is the list of block devices to be used by the container.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].volumeDevices[]`

Description

`volumeDevice` describes a mapping of a raw block device within a container.

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	<code>devicePath</code> is the path inside of the container that the device will be mapped to.

Property	Type	Description
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

`.spec.workloads[].template.spec.initContainers[].volumeMounts`

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

`.spec.workloads[].template.spec.initContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath`

`name`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>).
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be

Property	Type	Description
		unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

`.spec.workloads[].template.spec.nodeSelector`

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: <https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

`object`

`.spec.workloads[].template.spec.os`

Description

Specifies the OS of the containers in the pod. Some pod and container fields are restricted if this is set. If the OS field is set to linux, the following fields must be unset: - securityContext.windowsOptions If the OS field is set to windows, following fields must be unset: - spec.hostPID - spec.hostIPC - spec.hostUsers - spec.securityContext.appArmorProfile - spec.securityContext.seLinuxOptions - spec.securityContext.seccompProfile - spec.securityContext.fsGroup - spec.securityContext.fsGroupChangePolicy - spec.securityContext.sysctls - spec.shareProcessNamespace - spec.securityContext.runAsUser - spec.securityContext.runAsGroup - spec.securityContext.supplementalGroups - spec.securityContext.supplementalGroupsPolicy - spec.containers[*].securityContext.appArmorProfile - spec.containers[*].securityContext.seLinuxOptions - spec.containers[*].securityContext.seccompProfile - spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readOnlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsUser - spec.containers[*].securityContext.runAsGroup

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration ↗ Clients should expect to handle additional values and treat unrecognized values in this field as os: null

`.spec.workloads[].template.spec.overhead`

Description

Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info:

<https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md>

Type

object

`.spec.workloads[].template.spec.readinessGates`

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type

array

`.spec.workloads[].template.spec.readinessGates[]`

Description

PodReadinessGate contains the reference to a pod condition

Type

object

Required

conditionType

Property	Type	Description
<code>conditionType</code>	<code>string</code>	ConditionType refers to a condition in the pod's condition list with matching type.

`.spec.workloads[].template.spec.resourceClaims`

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.

Type

`array`

`.spec.workloads[].template.spec.resourceClaims[]`

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod. It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name uniquely identifies this resource claim inside the pod. This must be a

Property	Type	Description
		DNS_LABEL.
<code>resourceClaimName</code>	<code>string</code>	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.
<code>resourceClaimTemplateName</code>	<code>string</code>	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in pod.status.resourceClaimStatuses. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

`.spec.workloads[].template.spec.schedulingGates`

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

array

`.spec.workloads[].template.spec.schedulingGates[]`

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the scheduling gate. Each scheduling gate must have a unique name field.

`.spec.workloads[].template.spec.securityContext`

Description

SecurityContext holds pod-level security attributes and common container settings.
Optional: Defaults to empty. See type description for default values of each field.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>fsGroup</code>	<code>integer</code>	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>fsGroupChangePolicy</code>	<code>string</code>	<code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and

Property	Type	Description
		"Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that

Property	Type	Description
		container. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
supplementalGroups	array	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the

Property	Type	Description
		supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.workloads[].template.spec.securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.workloads[].template.spec.securityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext.

If set in both `SecurityContext` and `PodSecurityContext`, the value specified in `SecurityContext` takes precedence for that container. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

`.spec.workloads[].template.spec.securityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.workloads[].template.spec.securityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when `spec.os.name` is windows.

Type

`array`

`.spec.workloads[].template.spec.securityContext.supplementalGroups[]`

Type

integer

`.spec.workloads[].template.spec.securityContext.sysctls`

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

`.spec.workloads[].template.spec.securityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.workloads[].template.spec.securityContext.windows

Options

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.

Property	Type	Description
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.workloads[].template.spec.tolerations`

Description

If specified, the pod's tolerations.

Type

`array`

`.spec.workloads[].template.spec.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.workloads[].template.spec.topologySpreadConstraints`

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[]`

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

object

Required

maxSkew

topologyKey

whenUnsatisfiable

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values

Property	Type	Description
		from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).
maxSkew	integer	MaxSkew describes the degree to which pods may be unevenly distributed. When <code>whenUnsatisfiable=DoNotSchedule</code>, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1).

Property	Type	Description
		if MaxSkew is 2, incoming pod can be scheduled onto any zone. When <code>whenUnsatisfiable=ScheduleAnyway</code> , it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.

Property	Type	Description
<code>minDomains</code>	<code>integer</code>	MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule. For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: zone1 zone2 zone3 P P P P P P The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.
<code>nodeAffinityPolicy</code>	<code>string</code>	NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are:

Property	Type	Description
		- Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. - Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations. If this value is nil, the behavior is equivalent to the Honor policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
nodeTaintsPolicy	string	NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: - Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. - Ignore: node taints are ignored. All nodes are included. If this value is nil, the behavior is equivalent to the Ignore policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
topologyKey	string	TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket.

Property	Type	Description
		We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of <code>nodeAffinityPolicy</code> and <code>nodeTaintsPolicy</code>. e.g. If <code>TopologyKey</code> is "kubernetes.io/hostname", each <code>Node</code> is a domain of that topology. And, if <code>TopologyKey</code> is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.
<code>whenUnsatisfiable</code>	<code>string</code>	<code>WhenUnsatisfiable</code> indicates how to deal with a pod if it doesn't satisfy the spread constraint. • <code>DoNotSchedule</code> (default) tells the scheduler not to schedule it. • <code>ScheduleAnyway</code> tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, <code>MaxSkew</code> is set to 1, and pods with the same <code>labelSelector</code> spread as 3/1/1: zone1 zone2 zone3 P P P P P If <code>WhenUnsatisfiable</code> is set to <code>DoNotSchedule</code>, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as <code>ActualSkew(2-1)</code> on zone2(zone3) satisfies <code>MaxSkew(1)</code>. In other words, the cluster can still be imbalanced, but scheduler won't make it <i>more</i> imbalanced. It's a required field.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector`

Description

LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.workloads[].template.spec.topologySpreadConstraints[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.workloads[].template.spec.topologySpreadConstraints[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).

Type

array

`.spec.workloads[].template.spec.topologySpreadConstraints[].matchLabelKeys[]`

Type

`string`

`.spec.workloads[].template.spec.volumes`

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

`array`

`.spec.workloads[].template.spec.volumes[]`

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

`object`

Required

`name`

Property	Type	Description
<code>awsElasticBlockStore</code>	<code>object</code>	<code>awsElasticBlockStore</code> represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. https://kubernetes.io/docs/concepts/storage/volumes/#aws-elastic-block-store

Property	Type	Description
<code>azureDisk</code>	<code>object</code>	azureDisk represents an Azure Data Disk mount on mount to the pod.
<code>azureFile</code>	<code>object</code>	azureFile represents an Azure File Service mount or mount to the pod.
<code>cephfs</code>	<code>object</code>	cephFS represents a Ceph FS mount on the host th lifetime
<code>cinder</code>	<code>object</code>	cinder represents a cinder volume attached and mot machine. More info: https://examples.k8s.io/mysql-c
<code>configMap</code>	<code>object</code>	configMap represents a configMap that should popu
<code>csi</code>	<code>object</code>	csi (Container Storage Interface) represents ephem handled by certain external CSI drivers (Beta feature
<code>downwardAPI</code>	<code>object</code>	downwardAPI represents downward API about the p populate this volume

Property	Type	Description
<code>emptyDir</code>	<code>object</code>	<code>emptyDir</code> represents a temporary directory that shares its life cycle with the Pod (the volume is present as long as the Pod is running and is deleted when the Pod is terminated: either by a pod termination or by a node restart). More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
<code>ephemeral</code>	<code>object</code>	<code>ephemeral</code> represents a volume that is handled by a <code>PersistentVolumeClaim</code> and is only present while the pod is running. It has the same lifecycle as the pod: the volume is created before the pod starts, and deleted when the pod is removed or the node is restarted without a restart policy. Use this if: a) the volume is only needed while the pod is running, b) the volume is not needed for restoring from snapshot or capacity reservation, c) the storage driver is specified through a <code>storageClassName</code> and the storage driver supports dynamic volume provisioning. <code>ephemeral</code> is a shorthand for a <code>PersistentVolumeClaim</code> and a <code>Volume</code> of type <code>EmptyDir</code> or <code>HostPath</code> (see <code>EphemeralVolumeSource</code> for more information on the connection between this volume type and <code>PersistentVolumeClaim</code>). Use <code>PersistentVolumeClaim</code> or one of the vendor-specific <code>Volume</code> types for volumes that persist for longer than the lifecycle of a pod. Use CSI for light-weight local ephemeral volumes if it is not meant to be used that way - see the documentation for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.
<code>fc</code>	<code>object</code>	<code>fc</code> represents a Fibre Channel resource that is attached to a storage machine and then exposed to the pod.

Property	Type	Description
<code>flexVolume</code>	<code>object</code>	<code>flexVolume</code> represents a generic volume resource that is provisioned/attached using an exec based plugin.
<code>flocker</code>	<code>object</code>	<code>flocker</code> represents a Flocker volume attached to a kubelet. This depends on the Flocker control service being running.
<code>gcePersistentDisk</code>	<code>object</code>	<code>gcePersistentDisk</code> represents a GCE Disk resource that is provisioned on kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes/#gce-persistent-disk
<code>gitRepo</code>	<code>object</code>	<code>gitRepo</code> represents a git repository at a particular revision. <code>GitRepo</code> is deprecated. To provision a container with a git repo, use an <code>EmptyDir</code> into an <code>InitContainer</code> that clones the repo and then expose the <code>EmptyDir</code> into the Pod's container.
<code>glusterfs</code>	<code>object</code>	<code>glusterfs</code> represents a Glusterfs mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/glusterfs/
<code>hostPath</code>	<code>object</code>	<code>hostPath</code> represents a pre-existing file or directory on the host that is directly exposed to the container. This is generally used for logging agents or other privileged things that are allowed to see the host's file system. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes/#host-path

Property	Type	Description
<code>image</code>	<code>object</code>	image represents an OCI object (a container image mounted on the kubelet's host machine. The volume startup depending on which PullPolicy value is provided. • Always: the kubelet always attempts to pull the reference, even if it already exists locally. Container creation will fail if the pull fails. • Never: the kubelet never pulls the reference and only uses whatever image or artifact is present. Container creation will fail if the reference is not present. • IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and pull fails. The volume gets re-resolved if the pod gets deleted which means that new remote content will become available. Failure to resolve or pull the image during pod startup from starting and may add significant latency. Failure to pull will result in a normal volume backoff and will be reported on the pod status message. The types of objects that may be mounted are defined by the container runtime implementation on the host. The minimum must include all valid types supported by the container runtime. The OCI object gets mounted in a single direct layer (spec.containers[].volumeMounts.mountPath) by the container runtime in the same way as for container images. The volume is mounted read-only (ro) and non-executable files (noexec) mounts for containers are not supported (spec.containers[].volumeMounts.subpath). The field spec.securityContext.fsGroupChangePolicy has no effect on this type.

Property	Type	Description
<code>iscsi</code>	<code>object</code>	iscsi represents an ISCSI Disk resource that is attached to the host machine and then exposed to the pod. More info: https://examples.k8s.io/volumes/iscsi/README.md
<code>name</code>	<code>string</code>	name of the volume. Must be a DNS_LABEL and unique within the namespace. More info: https://kubernetes.io/docs/concepts/overview/objects/names/#names
<code>nfs</code>	<code>object</code>	nfs represents an NFS mount on the host that shares a pod-readable directory. More info: https://kubernetes.io/docs/concepts/storage/volumes/#nfs
<code>persistentVolumeClaim</code>	<code>object</code>	persistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes/#persistentvolumeclaims
<code>photonPersistentDisk</code>	<code>object</code>	photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine
<code>portworxVolume</code>	<code>object</code>	portworxVolume represents a portworx volume attached and mounted on kubelets host machine
<code>projected</code>	<code>object</code>	projected items for all in one resources secrets, configmaps, downward API

Property	Type	Description
<code>quobyte</code>	<code>object</code>	quobyte represents a Quobyte mount on the host throughout the pod's lifetime
<code>rbd</code>	<code>object</code>	rbd represents a Rados Block Device mount on the host throughout the pod's lifetime. More info: https://examples.k8s.io/volumes/rbd/README.md
<code>scaleIO</code>	<code>object</code>	scaleIO represents a ScaleIO persistent volume attached to the pod. More info: https://examples.k8s.io/volumes/scaleio/README.md
<code>secret</code>	<code>object</code>	secret represents a secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret
<code>storageos</code>	<code>object</code>	storageOS represents a StorageOS volume attached to the pod. More info: https://examples.k8s.io/volumes/storageos/README.md
<code>vsphereVolume</code>	<code>object</code>	vsphereVolume represents a vSphere volume attached to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#vspherevolume

`.spec.workloads[].template.spec.volumes[].awsElasticBlockStore`

Description

awsElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
partition	integer	partition is the partition in the volume that you want to mount. If not specified, the default is to mount by volume name. Examples: For volume "/dev/sda1", you specify the partition as "1". Similarly, the volume partition for "/dev/sda" is "0" (or you can leave the property empty).
readOnly	boolean	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

`.spec.workloads[].template.spec.volumes[].azureDisk`

Description

azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

object

Required

diskName

diskURI

Property	Type	Description
<code>cachingMode</code>	<code>string</code>	cachingMode is the Host Caching mode: None, Read Only, Read Write.
<code>diskName</code>	<code>string</code>	diskName is the Name of the data disk in the blob storage
<code>diskURI</code>	<code>string</code>	diskURI is the URI of data disk in the blob storage
<code>fsType</code>	<code>string</code>	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>kind</code>	<code>string</code>	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

`.spec.workloads[].template.spec.volumes[].azureFile`

Description

azureFile represents an Azure File Service mount on the host and bind mount to the pod.

Type

`object`

Required

`secretName`

`shareName`

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>secretName</code>	<code>string</code>	secretName is the name of secret that contains Azure Storage Account Name and Key
<code>shareName</code>	<code>string</code>	shareName is the azure share Name

`.spec.workloads[].template.spec.volumes[].cephfs`

Description

cephFS represents a Ceph FS mount on the host that shares a pod's lifetime

Type

object

Required

monitors

Property	Type	Description
<code>monitors</code>	<code>array</code>	<code>monitors</code> is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>path</code>	<code>string</code>	<code>path</code> is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>secretFile</code>	<code>string</code>	<code>secretFile</code> is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

Property	Type	Description
<code>secretRef</code>	<code>object</code>	secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>user</code>	<code>string</code>	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

`.spec.workloads[].template.spec.volumes[].cephfs.monitors`

Description

monitors is Required: Monitors is a collection of Ceph monitors More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

`array`

`.spec.workloads[].template.spec.volumes[].cephfs.monitors[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].cephfs.secretRef`

Description

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.workloads[].template.spec.volumes[].cinder

Description

cinder represents a cinder volume attached and mounted on kubelets host machine. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

Type

object

Required

volumeID

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code>. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is optional: points to a secret object containing parameters used to connect to OpenStack.
<code>volumeID</code>	<code>string</code>	<code>volumeID</code> used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

`.spec.workloads[].template.spec.volumes[].cinder.secretRef`

Description

`secretRef` is optional: points to a secret object containing parameters used to connect to OpenStack.

Type

`object`

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.workloads[].template.spec.volumes[].configMap

Description

configMap represents a configMap that should populate this volume

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified

Property	Type	Description
		paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

`.spec.workloads[].template.spec.volumes[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].csi`

Description

csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Type

object

Required

driver

Property	Type	Description
driver	string	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
fsType	string	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
nodePublishSecretRef	object	nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.
readOnly	boolean	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).

Property	Type	Description
<code>volumeAttributes</code>	<code>object</code>	<code>volumeAttributes</code> stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

`.spec.workloads[].template.spec.volumes[].csi.nodePublishSecretRef`

Description

`nodePublishSecretRef` is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI `NodePublishVolume` and `NodeUnpublishVolume` calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.spec.workloads[].template.spec.volumes[].csi.volumeAttributes`

Description

volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

object

`.spec.workloads[].template.spec.volumes[].downwardAPI`

Description

downwardAPI represents downward API about the pod that should populate this volume

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	Items is a list of downward API volume file

`.spec.workloads[].template.spec.volumes[].downwardAPI.items`

Description

Items is a list of downward API volume file

Type

array

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
<code>fieldRef</code>	<code>object</code>	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
<code>mode</code>	<code>integer</code>	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
<code>path</code>	<code>string</code>	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.volumes[].downwardAPI.items[].resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>string</code>	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.workloads[].template.spec.volumes[].emptyDir`

Description

emptyDir represents a temporary directory that shares a pod's lifetime. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#emptydir>

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
sizeLimit		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.workloads[].template.spec.volumes[].ephemeral

Description

ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see

EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual pod. Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.

Type

object

Property	Type	Description
volumeClaimTemplate	object	Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be <code><pod name>-<volume name></code> where <code><volume name></code> is the name from the <code>PodSpec.Volumes</code> array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will <i>not</i> be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created.

Property	Type	Description
		Required, must not be nil.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate`

Description

Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be `<pod name>-<volume name>` where `<volume name>` is the name from the `PodSpec.Volumes` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

Type

object

Required

spec

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code>	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.

Property	Type	Description
<code>spec</code>	<code>object</code>	The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec`

Description

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

Type

`object`

Property	Type	Description
<code>accessModes</code>	<code>array</code>	accessModes contains the desired access mode volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#access-modes-1
<code>dataSource</code>	<code>object</code>	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can support the specified data source, it will create a new

Property	Type	Description
		volume based on the contents of the specific source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If dataSourceRef.namespace is specified, then dataSourceRef contents will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-core API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the name of the specified object matches some installed volume populator or dynamic provisioner. This field will have the functionality of the dataSource field and as such, if both fields are non-empty, they must have the same value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: • While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves them.

Property	Type	Description
		values, and generates an error if a disallowe value is specified. While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
resources	object	resources represents the minimum resources th volume should have. If RecoverVolumeExpansionFailure feature is ena users are allowed to specify resource requirem that are lower than previous value but must still higher than capacity recorded in the status field claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources
selector	object	selector is a label query over volumes to consid binding.
storageClassName	string	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that the <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string value is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClassName</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ [↗] (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]`

Type

string

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef`

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.

(Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:
<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector`

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.workloads[].template.spec.volumes[].fc`

Description

fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Type

`object`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>lun</code>	<code>integer</code>	lun is Optional: FC target lun number
<code>readOnly</code>	<code>boolean</code>	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in

Property	Type	Description
		VolumeMounts.
<code>targetWWNs</code>	<code>array</code>	targetWWNs is Optional: FC target worldwide names (WWNs)
<code>wwids</code>	<code>array</code>	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

`.spec.workloads[].template.spec.volumes[].fc.targetWWNs`

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

`array`

`.spec.workloads[].template.spec.volumes[].fc.targetWWNs`
`[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].fc.wwids`

Description

wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

Type

`array`

`.spec.workloads[].template.spec.volumes[].fc.wwid[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].flexVolume`

Description

flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type

`object`

Required

`driver`

Property	Type	Description
<code>driver</code>	<code>string</code>	driver is the name of the driver to use for this volume.
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
<code>options</code>	<code>object</code>	options is Optional: this field holds extra command options if any.

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> is Optional: defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code>.
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is Optional: <code>secretRef</code> is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

`.spec.workloads[].template.spec.volumes[].flexVolume.options`

Description

`options` is Optional: this field holds extra command options if any.

Type

`object`

`.spec.workloads[].template.spec.volumes[].flexVolume.secretRef`

Description

`secretRef` is Optional: `secretRef` is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified.

If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].flocker`

Description

flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Type

`object`

Property	Type	Description
<code>datasetName</code>	<code>string</code>	datasetName is Name of the dataset stored as metadata - > name on the dataset for Flocker should be considered as deprecated
<code>datasetUUID</code>	<code>string</code>	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

`.spec.workloads[].template.spec.volumes[].gcePersistentDisk`

Description

`gcePersistentDisk` represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

Type

object

Required

pdName

Property	Type	Description
<code>fsType</code>	string	<code>fsType</code> is filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
<code>partition</code>	integer	<code>partition</code> is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
<code>pdName</code>	string	<code>pdName</code> is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> Defaults to <code>false</code>. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis

`.spec.workloads[].template.spec.volumes[].gitRepo`

Description

`gitRepo` represents a git repository at a particular revision. DEPRECATED: `GitRepo` is deprecated. To provision a container with a git repo, mount an `EmptyDir` into an `InitContainer` that clones the repo using `git`, then mount the `EmptyDir` into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
<code>directory</code>	<code>string</code>	<code>directory</code> is the target directory name. Must not contain or start with <code>'..'</code>. If <code>'.'</code> is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
<code>repository</code>	<code>string</code>	<code>repository</code> is the URL

Property	Type	Description
<code>revision</code>	<code>string</code>	revision is the commit hash for the specified revision.

`.spec.workloads[].template.spec.volumes[].glusterfs`

Description

glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/glusterfs/README.md>

Type

`object`

Required

`endpoints`

`path`

Property	Type	Description
<code>endpoints</code>	<code>string</code>	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
<code>path</code>	<code>string</code>	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

`.spec.workloads[].template.spec.volumes[].hostPath`

Description

`hostPath` represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

`.spec.workloads[].template.spec.volumes[].image`

Description

image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided:

- Always: the kubelet always attempts to pull the reference. Container creation will fail if the pull fails.
- Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present.
- IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (`spec.containers[*].volumeMounts.mountPath`) by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (`spec.containers[*].volumeMounts.subpath`). The field `spec.securityContext.fsGroupChangePolicy` has no effect on this volume type.

Type

object

Property	Type	Description
<code>pullPolicy</code>	<code>string</code>	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.
<code>reference</code>	<code>string</code>	Required: Image or artifact reference to be used. Behaves in the same way as <code>pod.spec.containers[*].image</code>. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

`.spec.workloads[].template.spec.volumes[].iscsi`

Description

iscsi represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://examples.k8s.io/volumes/iscsi/README.md>

Type

`object`

Required

`iqn`

`lun`

`targetPortal`

Property	Type	Description
<code>chapAuthDiscovery</code>	<code>boolean</code>	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
<code>chapAuthSession</code>	<code>boolean</code>	chapAuthSession defines whether support iSCSI Session CHAP authentication
<code>fsType</code>	<code>string</code>	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
<code>initiatorName</code>	<code>string</code>	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
<code>iqn</code>	<code>string</code>	iqn is the target iSCSI Qualified Name.
<code>iscsiInterface</code>	<code>string</code>	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
<code>lun</code>	<code>integer</code>	lun represents iSCSI Target Lun number.

Property	Type	Description
<code>portals</code>	<code>array</code>	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
<code>secretRef</code>	<code>object</code>	secretRef is the CHAP Secret for iSCSI target and initiators authentication
<code>targetPortal</code>	<code>string</code>	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

`.spec.workloads[].template.spec.volumes[].iscsi.portals`

Description

portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

Type

`array`

`.spec.workloads[].template.spec.volumes[].iscsi.portals[]`

Type

`string`

`.spec.workloads[].template.spec.volumes[].iscsi.secretRef`

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].nfs`

Description

nfs represents an NFS mount on the host that shares a pod's lifetime More info:
<https://kubernetes.io/docs/concepts/storage/volumes#nfs>

Type

object

Required

`path`

`server`

Property	Type	Description
<code>path</code>	<code>string</code>	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
<code>server</code>	<code>string</code>	<code>server</code> is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

`.spec.workloads[].template.spec.volumes[].persistentVolumeClaim`

Description

`persistentVolumeClaimVolumeSource` represents a reference to a `PersistentVolumeClaim` in the same namespace. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

`object`

Required

`claimName`

Property	Type	Description
<code>claimName</code>	<code>string</code>	<code>claimName</code> is the name of a <code>PersistentVolumeClaim</code> in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> Will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> . Default <code>false</code> .

`.spec.workloads[].template.spec.volumes[].photonPersistentDisk`

Description

`photonPersistentDisk` represents a `PhotonController` persistent disk attached and mounted on kubelets host machine

Type

`object`

Required

`pdID`

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>pdID</code>	<code>string</code>	<code>pdID</code> is the ID that identifies Photon Controller persistent disk

`.spec.workloads[].template.spec.volumes[].portworxVolume`

Description

portworxVolume represents a portworx volume attached and mounted on kubelets host machine

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

.spec.workloads[].template.spec.volumes[].projected

Description

projected items for all in one resources secrets, configmaps, and downward API

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	<code>defaultMode</code> are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like <code>fsGroup</code> , and the result can be other mode bits set.
<code>sources</code>	<code>array</code>	<code>sources</code> is the list of volume projections. Each entry in this list handles one source.

`.spec.workloads[].template.spec.volumes[].projected.sources`

Description

`sources` is the list of volume projections. Each entry in this list handles one source.

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[]`

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

`object`

Property	Type	Description
<code>clusterTrustBundle</code>	<code>object</code>	ClusterTrustBundle allows a pod to access the <code>.spec.trustBundle</code> field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.
<code>configMap</code>	<code>object</code>	configMap information about the configMap data to project
<code>downwardAPI</code>	<code>object</code>	downwardAPI information about the downwardAPI data to project
<code>secret</code>	<code>object</code>	secret information about the secret data to project

Property	Type	Description
<code>serviceAccountToken</code>	<code>object</code>	serviceAccountToken is information about the serviceAccountToken data to project

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle`

Description

ClusterTrustBundle allows a pod to access the `.spec.trustBundle` field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.`

Type

`object`

Required

`path`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Property	Type	Description
<code>name</code>	<code>string</code>	Select a single ClusterTrustBundle by object name. Mutually-exclusive with <code>signerName</code> and <code>labelSelector</code> .
<code>optional</code>	<code>boolean</code>	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using <code>name</code> , then the named ClusterTrustBundle is allowed not to exist. If using <code>signerName</code> , then the combination of <code>signerName</code> and <code>labelSelector</code> is allowed to match zero ClusterTrustBundles.
<code>path</code>	<code>string</code>	Relative path from the volume root to write the bundle.
<code>signerName</code>	<code>string</code>	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with <code>name</code> . The contents of all selected ClusterTrustBundles will be unified and deduplicated.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector`

Description

Select all ClusterTrustBundles that match this label selector. Only has effect if `signerName` is set. Mutually-exclusive with `name`. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[ ]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]`

Type

string

.spec.workloads[].template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.workloads[].template.spec.volumes[].projected.sources[].configMap

Description

configMap information about the configMap data to project

Type

object

Property	Type	Description
<code>items</code>	<code>array</code>	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
<code>optional</code>	<code>boolean</code>	optional specify whether the ConfigMap or its keys must be defined

`.spec.workloads[].template.spec.volumes[].projected.sources[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will

error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
<code>path</code>	<code>string</code>	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI`

Description

downwardAPI information about the downwardAPI data to project

Type

`object`

Property	Type	Description
<code>items</code>	<code>array</code>	Items is a list of DownwardAPIVolume file

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items`

Description

Items is a list of DownwardAPIVolume file

Type

`array`

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
<code>fieldRef</code>	<code>object</code>	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
<code>mode</code>	<code>integer</code>	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>path</code>	<code>string</code>	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory,

Property	Type	Description
		requests.cpu and requests.memory) are currently supported.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

object

Required

fieldPath

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.workloads[].template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor		Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

.spec.workloads[].template.spec.volumes[].projected.sources[].secret

Description

secret information about the secret data to project

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified

Property	Type	Description
		paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	optional field specify whether the Secret or its key must be defined

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.workloads[].template.spec.volumes[].projected.sources[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.workloads[].template.spec.volumes[].projected.sources[].serviceAccountToken`

Description

serviceAccountToken is information about the serviceAccountToken data to project

Type

object

Required

path

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours.Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

.spec.workloads[].template.spec.volumes[].quobyte

Description

quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Type

object

Required

registry

volume

Property	Type	Description
group	string	group to map volume access to Default is no group
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user
volume	string	volume is a string that references an already created Quobyte volume by name.

.spec.workloads[].template.spec.volumes[].rbd

Description

rbd represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/rbd/README.md>

Type

object

Required

image

monitors

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Property	Type	Description
<code>monitors</code>	<code>array</code>	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
<code>pool</code>	<code>string</code>	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code>. Defaults to <code>false</code>. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is name of the authentication secret for <code>RBDUser</code>. If provided overrides <code>keyring</code>. Default is <code>nil</code>. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
<code>user</code>	<code>string</code>	<code>user</code> is the rados user name. Default is <code>admin</code>. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

`.spec.workloads[].template.spec.volumes[].rbd.monitors`

Description

monitors is a collection of Ceph monitors. More info:

<https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

array

`.spec.workloads[].template.spec.volumes[].rbd.monitors[]`

Type

string

`.spec.workloads[].template.spec.volumes[].rbd.secretRef`

Description

secretRef is name of the authentication secret for RBDUser. If provided overrides keyring.

Default is nil. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].scaleIO`

Description

scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Type

object

Required

gateway

secretRef

system

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
gateway	string	gateway is the host address of the ScaleIO API Gateway.
protectionDomain	string	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.
sslEnabled	boolean	sslEnabled Flag enable/disable SSL communication with Gateway, default false

Property	Type	Description
<code>storageMode</code>	<code>string</code>	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
<code>storagePool</code>	<code>string</code>	storagePool is the ScaleIO Storage Pool associated with the protection domain.
<code>system</code>	<code>string</code>	system is the name of the storage system as configured in ScaleIO.
<code>volumeName</code>	<code>string</code>	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

`.spec.workloads[].template.spec.volumes[].scaleIO.secretRef`

Description

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty.

Property	Type	Description
		Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].secret`

Description

secret represents a secret that should populate this volume. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#secret>

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume

Property	Type	Description
		setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>optional</code>	<code>boolean</code>	optional field specify whether the Secret or its keys must be defined
<code>secretName</code>	<code>string</code>	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

`.spec.workloads[].template.spec.volumes[].secret.items`

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

`array`

`.spec.workloads[].template.spec.volumes[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

`object`

Required

key

path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.workloads[].template.spec.volumes[].storageos

Description

storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.

Type

object

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
<code>volumeNamespace</code>	<code>string</code>	<code>volumeNamespace</code> specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set <code>VolumeName</code> to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

`.spec.workloads[].template.spec.volumes[].storageos.secretRef`

Description

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.workloads[].template.spec.volumes[].vsphereVolume`

Description

vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Type

object

Required

`volumePath`

Property	Type	Description
<code>fsType</code>	<code>string</code>	<code>fsType</code> is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>storagePolicyID</code>	<code>string</code>	<code>storagePolicyID</code> is the storage Policy Based Management (SPBM) profile ID associated with the <code>StoragePolicyName</code> .
<code>storagePolicyName</code>	<code>string</code>	<code>storagePolicyName</code> is the storage Policy Based Management (SPBM) profile name.
<code>volumePath</code>	<code>string</code>	<code>volumePath</code> is the path that identifies vSphere volume vmrk

.status

Description

`ComponentStatus` defines the observed state of Connector

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	<code>Annotations</code> is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is

Property	Type	Description
		roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See:
<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
status	string	Status of the condition, one of True, False, Unknown.

Property	Type	Description
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsoci`
 - **DELETE** : delete collection of ConnectorsOCI
 - **GET** : list objects of kind ConnectorsOCI
 - **POST** : create a new ConnectorsOCI
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsoci/{name}`
 - **DELETE** : delete the specified ConnectorsOCI
 - **GET** : read the specified ConnectorsOCI
 - **PATCH** : partially update the specified ConnectorsOCI
 - **PUT** : replace the specified ConnectorsOCI
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsoci/{name}/status`
 - **GET** : read status of the specified ConnectorsOCI
 - **PATCH** : partially update status of the specified ConnectorsOCI
 - **PUT** : replace status of the specified ConnectorsOCI

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsoci`

HTTP method

DELETE

Description

delete collection of ConnectorsOCI

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ConnectorsOCI

HTTP responses

HTTP code	Response body
200 - OK	ConnectorsOCICollection schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ConnectorsOCI

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsOCI</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsOCI</code> schema
201 - Created	<code>ConnectorsOCI</code> schema
202 - Accepted	<code>ConnectorsOCI</code> schema
401 - Unauthorized	Empty

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsocis/{name}

HTTP method

DELETE

Description

delete the specified ConnectorsOCI

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ConnectorsOCI

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsOCI</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ConnectorsOCI

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsOCI</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ConnectorsOCI

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsOCI</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsOCI</code> schema
201 - Created	<code>ConnectorsOCI</code> schema
401 - Unauthorized	Empty

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsocis/{name}/status`

HTTP method

GET

Description

read status of the specified ConnectorsOCI

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsOCI</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ConnectorsOCI

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsOCI</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectorsOCI

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsOCI</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsOCI</code> schema
201 - Created	<code>ConnectorsOCI</code> schema
401 - Unauthorized	Empty