Workbench APIs

Workspace Kind [kubeflow.org/ Workspace [kubeflow.org/v1beta1]

Workspace Kind [kubeflow.org/v1beta1]

Description

WorkspaceKind is the Schema for the WorkspaceKinds API

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	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 ↗

Property	Type	Description
`kind`	`string`	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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta` ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	WorkspaceKindSpec defines the desired state of WorkspaceKind
`status`	`object`	WorkspaceKindStatus defines the observed state of WorkspaceKind

.spec

Description

WorkspaceKindSpec defines the desired state of WorkspaceKind

Type

`object`

Required

`podTemplate` `spawner`

Property	Type	Description
`podTemplate`	`object`	podTemplate is the PodTemplate used to spawn Pods to run Workspaces of this WorkspaceKind
`spawner`	`object`	spawner config determines how the WorkspaceKind is displayed in the Workspace Spawner UI

.spec.podTemplate

Description

podTemplate is the PodTemplate used to spawn Pods to run Workspaces of this WorkspaceKind

Type

`object`

Required

`options` `serviceAccount` `volumeMounts`

Property	Type	Description
`containerSecurityContext`	`object`	container security context for Workspace Pods (MUTABLE)
`culling`	`object`	culling configs for pausing inactive Workspaces (MUTABLE)
`extraEnv`	`array`	environment variables for Workspace Pods (MUTABLE)

Property	Type	Description
		• the following go template functions are available: • `httpPathPrefix(portId string)`: returns the HTTP path prefix of the specified port
`extraVolumeMounts`	`array`	extra volume mounts for Workspace Pods (MUTABLE)
`extraVolumes`	`array`	extra volumes for Workspace Pods (MUTABLE)
`httpProxy`	`object`	http proxy configs (MUTABLE)
`options`	`object`	options are the user-selectable fields, they determine the PodSpec of the Workspace
`podMetadata`	`object`	metadata for Workspace Pods (MUTABLE)
`probes`	`object`	standard probes to determine Container health (MUTABLE)
`securityContext`	`object`	security context for Workspace Pods (MUTABLE)

Property	Type	Description
`serviceAccount`	`object`	service account configs for Workspace Pods
`volumeMounts`	`object`	volume mount paths

.spec.podTemplate.containerSecurityContext

Description

container security context for Workspace Pods (MUTABLE)

Type

`object`

Property	Type	Description
`allowPrivilegeEscalation`	`boolean`	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
`appArmorProfile`	`object`	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note

Property	Type	Description
		that this field cannot be set when spec.os.name is windows.
`capabilities`	`object`	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.
`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.

Property	Type	Description
`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 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.

Property	Type	Description
`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 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.podTemplate.containerSecurityContext.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.podTemplate.containerSecurityContext.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
`drop`	`array`	Removed capabilities

.spec.podTemplate.containerSecurityContext.capabilities.add

Description

Added capabilities

Type

`array`

.spec.podTemplate.containerSecurityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.podTemplate.containerSecurityContext.capabilities.drop

Description

Removed capabilities

Type

`array`

.spec.podTemplate.containerSecurityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.podTemplate.containerSecurityContext.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
`user`	`string`	User is a SELinux user label that applies to the container.

.spec.podTemplate.containerSecurityContext.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
`localhostProfile`	`string`	localhostProfile 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.
`type`	`string`	type 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.podTemplate.containerSecurityContext.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
`gmsaCredentialSpec`	`string`	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 GMSACredentialSpecName field.
`gmsaCredentialSpecName`	`string`	GMSACredentialSpecName is the name of the GMSA credential spec to use.
`hostProcess`	`boolean`	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.podTemplate.culling

Description

culling configs for pausing inactive Workspaces (MUTABLE)

Type

`object`

Required

`activityProbe`

Property	Type	Description
`activityProbe`	`object`	the probe used to determine if the Workspace is active

Property	Type	Description
`enabled`	`boolean`	if the culling feature is enabled
`maxInactiveSeconds`	`integer`	the maximum number of seconds a Workspace can be inactive

.spec.podTemplate.culling.activityProbe

Description

the probe used to determine if the Workspace is active

Type

`object`

Property	Type	Description
`exec`	`object`	a shell command probe • if the Workspace had activity in the last 60 seconds this command should return status 0, otherwise it should return status 1
`jupyter`	`object`	a Jupyter-specific probe • will poll the `/api/status` endpoint of the Jupyter API, and use the `last_activity` field • note, users need to be careful that their other probes don't trigger a "last_activity" update e.g. they should only check the health of Jupyter using the `/api/status` endpoint

.spec.podTemplate.culling.activityProbe.exec

Description

a shell command probe - if the Workspace had activity in the last 60 seconds this command should return status 0, otherwise it should return status 1

Type

`object`

Required

`command`

Property	Type	Description
`command`	`array`	the command to run

.spec.podTemplate.culling.activityProbe.exec.command

Description

the command to run

Type

`array`

.spec.podTemplate.culling.activityProbe.exec.command[]

Type

`string`

.spec.podTemplate.culling.activityProbe.jupyter

Description

a Jupyter-specific probe - will poll the `/api/status` endpoint of the Jupyter API, and use the `last_activity` field - note, users need to be careful that their other probes don't trigger a

"last_activity" update e.g. they should only check the health of Jupyter using the `/api/status` endpoint

Type

`object`

Required

`lastActivity`

Property	Type	Description
`lastActivity`	`boolean`	if the Jupyter-specific probe is enabled

.spec.podTemplate.extraEnv

Description

environment variables for Workspace Pods (MUTABLE) - the following go template functions are available: - `httpPathPrefix(portId string)`: returns the HTTP path prefix of the specified port

Type

`array`

.spec.podTemplate.extraEnv[]

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.podTemplate.extraEnv[].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.

Property	Type	Description
`fieldRef`	`object`	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.
`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.podTemplate.extraEnv[].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.podTemplate.extraEnv[].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
`apiVersion`	`string`	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
`fieldPath`	`string`	Path of the field to select in the specified API version.

.spec.podTemplate.extraEnv[].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
`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.podTemplate.extraEnv[].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.podTemplate.extraVolumeMounts

Description

extra volume mounts for Workspace Pods (MUTABLE)

Type

`array`

.spec.podTemplate.extraVolumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath` `name`

Property	Type	Description
`mountPath`	`string`	Path within the container at which the volume should be mounted. Must not contain ':'.
`mountPropagation`	`string`	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).
`name`	`string`	This must match the Name of a Volume.
`readOnly`	`boolean`	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
`recursiveReadOnly`	`string`	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified.

Property	Type	Description
		If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, 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 IfPossible or Enabled, MountPropagation must be set to None (or be 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.podTemplate.extraVolumes

Description

extra volumes for Workspace Pods (MUTABLE)

Type

`array`

.spec.podTemplate.extraVolumes[]

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
`awsElasticBlockStore`	`object`	awsElasticBlockStore represents an AWS Disk reso a kubelet's host machine and then exposed to the p https://kubernetes.io/docs/concepts/storage/volume ↗
`azureDisk`	`object`	azureDisk represents an Azure Data Disk mount on mount to the pod.
`azureFile`	`object`	azureFile represents an Azure File Service mount o mount to the pod.

Property	Type	Description
`cephfs`	`object`	cephFS represents a Ceph FS mount on the host th lifetime
`cinder`	`object`	cinder represents a cinder volume attached and mou machine. More info: https://examples.k8s.io/mysql-c ↗
`configMap`	`object`	configMap represents a configMap that should popu
`csi`	`object`	csi (Container Storage Interface) represents epheme handled by certain external CSI drivers (Beta feature
`downwardAPI`	`object`	downwardAPI represents downward API about the p populate this volume
`emptyDir`	`object`	emptyDir represents a temporary directory that shar More info: https://kubernetes.io/docs/concepts/stora ↗
`ephemeral`	`object`	ephemeral represents a volume that is handled by a The volume's lifecycle is tied to the pod that defines before the pod starts, and deleted when the pod is r Use this if: a) the volume is only needed while the po normal volumes like restoring from snapshot or capa needed, c) the storage driver is specified through a s

Property	Type	Description
		the storage driver supports dynamic volume provisio PersistentVolumeClaim (see EphemeralVolumeSour information on the connection between this volume t PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-sp volumes that persist for longer than the lifecycle of a Use CSI for light-weight local ephemeral volumes if t meant to be used that way - see the documentation information. A pod can use both types of ephemeral volumes and at the same time.
`fc`	`object`	fc represents a Fibre Channel resource that is attach machine and then exposed to the pod.
`flexVolume`	`object`	flexVolume represents a generic volume resource th provisioned/attached using an exec based plugin.
`flocker`	`object`	flocker represents a Flocker volume attached to a ku This depends on the Flocker control service being ru
`gcePersistentDisk`	`object`	gcePersistentDisk represents a GCE Disk resource kubelet's host machine and then exposed to the pod https://kubernetes.io/docs/concepts/storage/volumes

Property	Type	Description
`gitRepo`	`object`	gitRepo represents a git repository at a particular rev GitRepo is deprecated. To provision a container with EmptyDir into an InitContainer that clones the repo u the EmptyDir into the Pod's container.
`glusterfs`	`object`	glusterfs represents a Glusterfs mount on the host th lifetime. More info: https://examples.k8s.io/volumes/g ↗
`hostPath`	`object`	hostPath represents a pre-existing file or directory o that is directly exposed to the container. This is gene agents or other privileged things that are allowed to Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes
`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 provi • Always: the kubelet always attempts to pull the re creation will fail If the pull fails. • Never: the kubelet never pulls the reference and image or artifact. Container creation will fail if the present. • IfNotPresent: the kubelet pulls if the reference isr disk. Container creation will fail if the reference is pull fails. The volume gets re-resolved if the pod gets deleted means that new remote content will become availab

Property	Type	Description
		failure to resolve or pull the image during pod startup from starting and may add significant latency. Failure normal volume backoff and will be reported on the p message. The types of objects that may be mounted defined by the container runtime implementation on minimum must include all valid types supported by tl field. The OCI object gets mounted in a single direct (spec.containers[*].volumeMounts.mountPath) by me *layers in the same way as for container images. The mounted read-only (ro) and non-executable files (no mounts for containers are not supported (spec.containers[*].volumeMounts.subpath). The field spec.securityContext.fsGroupChangePolicy has no type.
`iscsi`	`object`	iscsi represents an ISCSI Disk resource that is attac 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 un More info: https://kubernetes.io/docs/concepts/overv objects/names/#names ↗
`nfs`	`object`	nfs represents an NFS mount on the host that share info: https://kubernetes.io/docs/concepts/storage/vol
`persistentVolumeClaim`	`object`	persistentVolumeClaimVolumeSource represents a r PersistentVolumeClaim in the same namespace. Mo

Property	Type	Description
		https://kubernetes.io/docs/concepts/storage/persiste volumes#persistentvolumeclaims ↗
`photonPersistentDisk`	`object`	photonPersistentDisk represents a PhotonController attached and mounted on kubelets host machine
`portworxVolume`	`object`	portworxVolume represents a portworx volume attac kubelets host machine
`projected`	`object`	projected items for all in one resources secrets, conf downward API
`quobyte`	`object`	quobyte represents a Quobyte mount on the host tha lifetime
`rbd`	`object`	rbd represents a Rados Block Device mount on the pod's lifetime. More info: https://examples.k8s.io/volumes/rbd/README.md ↗
`scaleIO`	`object`	scaleIO represents a ScaleIO persistent volume atta Kubernetes nodes.
`secret`	`object`	secret represents a secret that should populate this https://kubernetes.io/docs/concepts/storage/volumes

Property	Type	Description
`storageos`	`object`	storageOS represents a StorageOS volume attache Kubernetes nodes.
`vsphereVolume`	`object`	vsphereVolume represents a vSphere volume attach kubelets host machine

.spec.podTemplate.extraVolumes[].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

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	fsType is the filesystem type of the volume that you want to mou Ensure that the filesystem type is supported by the host operatir Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if u More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗
`partition`	`integer`	partition is the partition in the volume that you want to mount. If the default is to mount by volume name. Examples: For volume

Property	Type	Description
		you specify the partition as "1". Similarly, the volume partition fo is "0" (or you can leave the property empty).
`readOnly`	`boolean`	readOnly value true will force the readOnly setting in VolumeMo info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗
`volumeID`	`string`	volumeID is unique ID of the persistent disk resource in AWS (A EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗

.spec.podTemplate.extraVolumes[].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
`cachingMode`	`string`	cachingMode is the Host Caching mode: None, Read Only, Read Write.

Property	Type	Description
`diskName`	`string`	diskName is the Name of the data disk in the blob storage
`diskURI`	`string`	diskURI is the URI of data disk in the blob storage
`fsType`	`string`	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.
`kind`	`string`	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
`readOnly`	`boolean`	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

.spec.podTemplate.extraVolumes[].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
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretName`	`string`	secretName is the name of secret that contains Azure Storage Account Name and Key
`shareName`	`string`	shareName is the azure share Name

.spec.podTemplate.extraVolumes[].cephfs

Description

cephFS represents a Ceph FS mount on the host that shares a pod's lifetime

Type

`object`

Required

`monitors`

Property	Type	Description
`monitors`	`array`	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

Property	Type	Description
`path`	`string`	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
`readOnly`	`boolean`	readOnly 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 ↗
`secretFile`	`string`	secretFile 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 ↗
`secretRef`	`object`	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 ↗
`user`	`string`	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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].cephfs.monitors[]

Type

`string`

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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
`fsType`	`string`	fsType 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 ↗
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
`secretRef`	`object`	secretRef is optional: points to a secret object containing parameters used to connect to OpenStack.
`volumeID`	`string`	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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

Property	Type	Description
		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 '..'.
`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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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
`volumeAttributes`	`object`	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

.spec.podTemplate.extraVolumes[].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
`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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

`array`

.spec.podTemplate.extraVolumes[].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 '..'

Property	Type	Description
		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, requests.cpu and requests.memory) are currently supported.

.spec.podTemplate.extraVolumes[].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
`apiVersion`	`string`	Version of the schema the FieldPath is written in terms of, defaults to "v1".
`fieldPath`	`string`	Path of the field to select in the specified API version.

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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 `<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.

.spec.podTemplate.extraVolumes[].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.
`spec`	`object`	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.podTemplate.extraVolumes[].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
`accessModes`	`array`	accessModes contains the desired access mod volume should have. More info: https://kubernetes.io/docs/concepts/storage/per volumes#access-modes-1 ↗
`dataSource`	`object`	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 sup the specified data source, it will create a nev volume based on the contents of the specifie source. When the AnyVolumeDataSource fe gate is enabled, dataSource contents will be to dataSourceRef, and dataSourceRef conte be copied to dataSource when dataSourceRef.namespace is not specified. namespace is specified, then dataSourceRe not be copied to dataSource.

Property	Type	Description
`dataSourceRef`	`object`	dataSourceRef specifies the object from which t populate the volume with data, if a non-empty v is desired. This may be any object from a non-e API group (non core object) or a PersistentVolumeClaim object. When this field i specified, volume binding will only succeed if th of the specified object matches some installed v populator or dynamic provisioner. This field will the functionality of the dataSource field and as s both fields are non-empty, they must have the s value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to t same value automatically if one of them is empt the other is non-empty. When namespace is sp in dataSourceRef, dataSource isn't set to the sa value and must be empty. There are three impo differences between dataSource and dataSourc • While dataSource only allows two specific ty objects, dataSourceRef allows any non-core as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves a 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.

Property	Type	Description
`resources`	`object`	resources represents the minimum resources th volume should have. If RecoverVolumeExpansionFailure feature is ena users are allowed to specify resource requireme 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 ↗
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string value means t VolumeAttributesClass will be applied to the cla it's not allowed to reset this field to empty string is set. If unspecified and the PersistentVolumeC unbound, the default VolumeAttributesClass wil

Property	Type	Description
		by the persistentvolume controller if it exists. If t resource referred to by volumeAttributesClass d not exist, this PersistentVolumeClaim will be se Pending state, as reflected by the modifyVolum field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/ (Beta) Using this field requ the VolumeAttributesClass feature gate to be en (off by default).
`volumeMode`	`string`	volumeMode defines what type of volume is rec by the claim. Value of Filesystem is implied whe included in claim spec.
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]

Type

`string`

.spec.podTemplate.extraVolumes[].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
`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

.spec.podTemplate.extraVolumes[].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.

Property	Type	Description
`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.podTemplate.extraVolumes[].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
`limits`	`object`	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
`requests`	`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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].ephemeral.volumeClaimTemplate.spec.selector

Description

selector is a label query over volumes to consider for binding.

Type

`object`

Property	Type	Description
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.extraVolumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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
`fsType`	`string`	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.
`lun`	`integer`	lun is Optional: FC target lun number
`readOnly`	`boolean`	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`targetWWNs`	`array`	targetWWNs is Optional: FC target worldwide names (WWNs)
`wwids`	`array`	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

.spec.podTemplate.extraVolumes[].fc.targetWWNs

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

`array`

.spec.podTemplate.extraVolumes[].fc.targetWWNs[]

Type

`string`

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].fc.wwids[]

Type

`string`

.spec.podTemplate.extraVolumes[].flexVolume

Description

flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type

`object`

Required

`driver`

Property	Type	Description
`driver`	`string`	driver is the name of the driver to use for this volume.

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". The default filesystem depends on FlexVolume script.
`options`	`object`	options is Optional: this field holds extra command options if any.
`readOnly`	`boolean`	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretRef`	`object`	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.podTemplate.extraVolumes[].flexVolume.options

Description

options is Optional: this field holds extra command options if any.

Type

`object`

.spec.podTemplate.extraVolumes[].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
`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.podTemplate.extraVolumes[].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
`datasetName`	`string`	datasetName is Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated

Property	Type	Description
`datasetUUID`	`string`	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

.spec.podTemplate.extraVolumes[].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
`fsType`	`string`	fsType is filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operatir system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "e unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗
`partition`	`integer`	partition 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 info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗
`pdName`	`string`	pdName is unique name of the PD resource in GCE. Used to ide the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗
`readOnly`	`boolean`	readOnly here will force the ReadOnly setting in VolumeMounts Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗

.spec.podTemplate.extraVolumes[].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
`directory`	`string`	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be

Property	Type	Description
		the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
`repository`	`string`	repository is the URL
`revision`	`string`	revision is the commit hash for the specified revision.

.spec.podTemplate.extraVolumes[].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
`endpoints`	`string`	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗
`path`	`string`	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-

Property	Type	Description
		a-pod ↗
`readOnly`	`boolean`	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.podTemplate.extraVolumes[].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
`path`	`string`	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 ↗

Property	Type	Description
`type`	`string`	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗

.spec.podTemplate.extraVolumes[].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
`pullPolicy`	`string`	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.
`reference`	`string`	Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. 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.podTemplate.extraVolumes[].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
`chapAuthDiscovery`	`boolean`	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
`chapAuthSession`	`boolean`	chapAuthSession defines whether support iSCSI Sessior CHAP authentication
`fsType`	`string`	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supporte 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#isc ↗
`initiatorName`	`string`	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
`iqn`	`string`	iqn is the target iSCSI Qualified Name.
`iscsiInterface`	`string`	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
`lun`	`integer`	lun represents iSCSI Target Lun number.

Property	Type	Description
`portals`	`array`	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).
`readOnly`	`boolean`	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
`secretRef`	`object`	secretRef is the CHAP Secret for iSCSI target and initiato authentication
`targetPortal`	`string`	targetPortal is iSCSI Target Portal. The Portal is either ar IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].iscsi.portals[]

Type

`string`

.spec.podTemplate.extraVolumes[].iscsi.secretRef

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

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.podTemplate.extraVolumes[].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
`path`	`string`	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

Property	Type	Description
`readOnly`	`boolean`	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 ↗
`server`	`string`	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

.spec.podTemplate.extraVolumes[].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
`claimName`	`string`	claimName is the name of a PersistentVolumeClaim in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

Property	Type	Description
`readOnly`	`boolean`	readOnly Will force the ReadOnly setting in VolumeMounts. Default false.

.spec.podTemplate.extraVolumes[].photonPersistentDisk

Description

photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine

Type

`object`

Required

`pdID`

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". Implicitly inferred to be "ext4" if unspecified.
`pdID`	`string`	pdID is the ID that identifies Photon Controller persistent disk

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].projected

Description

projected items for all in one resources secrets, configmaps, and downward API

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	defaultMode 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 fsGroup, and the result can be other mode bits set.
`sources`	`array`	sources is the list of volume projections. Each entry in this list handles one source.

.spec.podTemplate.extraVolumes[].projected.sources

Description

sources is the list of volume projections. Each entry in this list handles one source.

Type

`array`

.spec.podTemplate.extraVolumes[].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
`clusterTrustBundle`	`object`	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.
`configMap`	`object`	configMap information about the configMap data to project
`downwardAPI`	`object`	downwardAPI information about the downwardAPI data to project
`secret`	`object`	secret information about the secret data to project

Property	Type	Description
`serviceAccountToken`	`object`	serviceAccountToken is information about the serviceAccountToken data to project

.spec.podTemplate.extraVolumes[].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
`labelSelector`	`object`	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
`name`	`string`	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
`optional`	`boolean`	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
`path`	`string`	Relative path from the volume root to write the bundle.
`signerName`	`string`	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

.spec.podTemplate.extraVolumes[].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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.extraVolumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].projected.sources[].configMap

Description

configMap information about the configMap data to project

Type

`object`

Property	Type	Description
`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 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.

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 ↗
`optional`	`boolean`	optional specify whether the ConfigMap or its keys must be defined

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.
`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.podTemplate.extraVolumes[].projected.sources[].downwardAPI

Description

downwardAPI information about the downwardAPI data to project

Type

`object`

Property	Type	Description
`items`	`array`	Items is a list of DownwardAPIVolume file

.spec.podTemplate.extraVolumes[].projected.sources[].downwardAPI.items

Description

Items is a list of DownwardAPIVolume file

Type

`array`

.spec.podTemplate.extraVolumes[].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.

Property	Type	Description
`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, requests.cpu and requests.memory) are currently supported.

.spec.podTemplate.extraVolumes[].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
`apiVersion`	`string`	Version of the schema the FieldPath is written in terms of, defaults to "v1".
`fieldPath`	`string`	Path of the field to select in the specified API version.

.spec.podTemplate.extraVolumes[].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"

Property	Type	Description
`resource`	`string`	Required: resource to select

.spec.podTemplate.extraVolumes[].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 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 ↗

Property	Type	Description
`optional`	`boolean`	optional field specify whether the Secret or its key must be defined

.spec.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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 serivceaccount user
`volume`	`string`	volume is a string that references an already created Quobyte volume by name.

.spec.podTemplate.extraVolumes[].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 ↗
`monitors`	`array`	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

Property	Type	Description
`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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].rbd.monitors[]

Type

`string`

.spec.podTemplate.extraVolumes[].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
`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.podTemplate.extraVolumes[].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
`storageMode`	`string`	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.

Property	Type	Description
`storagePool`	`string`	storagePool is the ScaleIO Storage Pool associated with the protection domain.
`system`	`string`	system is the name of the storage system as configured in ScaleIO.
`volumeName`	`string`	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

.spec.podTemplate.extraVolumes[].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
`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.podTemplate.extraVolumes[].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
`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 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 '..'.
`optional`	`boolean`	optional field specify whether the Secret or its keys must be defined

Property	Type	Description
`secretName`	`string`	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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].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.podTemplate.extraVolumes[].storageos

Description

storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.

Type

`object`

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". Implicitly inferred to be "ext4" if unspecified.
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretRef`	`object`	secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.
`volumeName`	`string`	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
`volumeNamespace`	`string`	volumeNamespace 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 VolumeName 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.podTemplate.extraVolumes[].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
`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.podTemplate.extraVolumes[].vsphereVolume

Description

vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Type

`object`

Required

`volumePath`

Property	Type	Description
`fsType`	`string`	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.

Property	Type	Description
`storagePolicyID`	`string`	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
`storagePolicyName`	`string`	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
`volumePath`	`string`	volumePath is the path that identifies vSphere volume vmdk

.spec.podTemplate.httpProxy

Description

http proxy configs (MUTABLE)

Type

`object`

Property	Type	Description
`removePathPrefix`	`boolean`	if the path prefix is stripped from incoming HTTP requests • if true, the '/workspace/{profile_name}/{workspace_nan path prefix is stripped from incoming requests, the application sees the request as if it was made to '/...' • this only works if the application serves RELATIVE URI its assets

Property	Type	Description
`requestHeaders`	`object`	header manipulation rules for incoming HTTP requests • sets the `spec.http[].headers.request` of the Istio VirtualService https://istio.io/latest/docs/reference/config/networking/virtual-service/#Headers-HeaderOperations ↗ • the following string templates are available: • `.PathPrefix`: the path prefix of the Workspace (e.g. '/workspace/{profile_name}/{workspace_name}/')

.spec.podTemplate.httpProxy.requestHeaders

Description

header manipulation rules for incoming HTTP requests - sets the `spec.http[].headers.request` of the Istio VirtualService https://istio.io/latest/docs/reference/config/networking/virtual-service/#Headers-HeaderOperations - the following string templates are available: - `.PathPrefix`: the path prefix of the Workspace (e.g. '/workspace/{profile_name}/{workspace_name}/')

Type

`object`

Property	Type	Description
`add`	`object`	append the given values to the headers specified by keys (will create a comma-separated list of values)
`remove`	`array`	remove the specified headers

Property	Type	Description
`set`	`object`	overwrite the headers specified by key with the given values

.spec.podTemplate.httpProxy.requestHeaders.add

Description

append the given values to the headers specified by keys (will create a comma-separated list of values)

Type

`object`

.spec.podTemplate.httpProxy.requestHeaders.remove

Description

remove the specified headers

Type

`array`

.spec.podTemplate.httpProxy.requestHeaders.remove[]

Type

`string`

.spec.podTemplate.httpProxy.requestHeaders.set

Description

overwrite the headers specified by key with the given values

Type

`object`

.spec.podTemplate.options

Description

options are the user-selectable fields, they determine the PodSpec of the Workspace

Type

`object`

Required

`imageConfig` `podConfig`

Property	Type	Description
`imageConfig`	`object`	imageConfig options
`podConfig`	`object`	podConfig options

.spec.podTemplate.options.imageConfig

Description

imageConfig options

Type

`object`

Required

`spawner` `values`

Property	Type	Description
`spawner`	`object`	spawner ui configs
`values`	`array`	the list of image configs that are available

.spec.podTemplate.options.imageConfig.spawner

Description

spawner ui configs

Type

`object`

Required

`default`

Property	Type	Description
`default`	`string`	the id of the default option • this will be selected by default in the spawner ui

.spec.podTemplate.options.imageConfig.values

Description

the list of image configs that are available

Type

`array`

.spec.podTemplate.options.imageConfig.values[]

Type

`object`

Required

`id` `spawner` `spec`

Property	Type	Description
`id`	`string`	the id of this image config
`redirect`	`object`	redirect configs
`spawner`	`object`	information for the spawner ui
`spec`	`object`	the spec of the image config

.spec.podTemplate.options.imageConfig.values[].redirect

Description

redirect configs

Type

`object`

Required

`to`

Property	Type	Description
`message`	`object`	information about the redirect
`to`	`string`	the id of the option to redirect to

.spec.podTemplate.options.imageConfig.values[].redirect.message

Description

information about the redirect

Type

`object`

Required

`level` `text`

Property	Type	Description
`level`	`string`	the importance level of the message
`text`	`string`	the text of the message to show

.spec.podTemplate.options.imageConfig.values[].spawner

Description

information for the spawner ui

Type

`object`

Required

`displayName`

Property	Type	Description
`description`	`string`	a description of the option
`displayName`	`string`	the display name of the option
`hidden`	`boolean`	if this option should be hidden from the Workspace Spawner UI
`labels`	`array`	labels for the option

.spec.podTemplate.options.imageConfig.values[].spawner.labels

Description

labels for the option

Type

`array`

.spec.podTemplate.options.imageConfig.values[].spawner.labels[]

Type

`object`

Required

`key` `value`

Property	Type	Description
`key`	`string`	the key of the label
`value`	`string`	the value of the label

.spec.podTemplate.options.imageConfig.values[].spec

Description

the spec of the image config

Type

`object`

Required

`image` `ports`

Property	Type	Description
`image`	`string`	the container image to use
`imagePullPolicy`	`string`	the pull policy for the container image
`ports`	`array`	ports that the container listens on

Property	Type	Description
		• if multiple ports are defined, the user will see multiple "Connect" buttons in a dropdown menu on the Workspace overview page

.spec.podTemplate.options.imageConfig.values[].spec.ports

Description

ports that the container listens on - if multiple ports are defined, the user will see multiple "Connect" buttons in a dropdown menu on the Workspace overview page

Type

`array`

.spec.podTemplate.options.imageConfig.values[].spec.ports[]

Type

`object`

Required

`displayName` `id` `port` `protocol`

Property	Type	Description
`displayName`	`string`	the display name of the port

Property	Type	Description
`id`	`string`	the id of the port • this is NOT used as the Container or Service port name, but as part of the HTTP path
`port`	`integer`	the port number
`protocol`	`string`	the protocol of the port

.spec.podTemplate.options.podConfig

Description

podConfig options

Type

`object`

Required

`spawner` `values`

Property	Type	Description
`spawner`	`object`	spawner ui configs
`values`	`array`	the list of pod configs that are available

.spec.podTemplate.options.podConfig.spawner

Description

spawner ui configs

Type

`object`

Required

`default`

Property	Type	Description
`default`	`string`	the id of the default option • this will be selected by default in the spawner ui

.spec.podTemplate.options.podConfig.values

Description

the list of pod configs that are available

Type

`array`

.spec.podTemplate.options.podConfig.values[]

Type

`object`

Required

`id` `spawner` `spec`

Property	Type	Description
`id`	`string`	the id of this pod config

Property	Type	Description
`redirect`	`object`	redirect configs
`spawner`	`object`	information for the spawner ui
`spec`	`object`	the spec of the pod config

.spec.podTemplate.options.podConfig.values[].redirect

Description

redirect configs

Type

`object`

Required

`to`

Property	Type	Description
`message`	`object`	information about the redirect
`to`	`string`	the id of the option to redirect to

.spec.podTemplate.options.podConfig.values[].redirect.message

Description

information about the redirect

Type

`object`

Required

`level` `text`

Property	Type	Description
`level`	`string`	the importance level of the message
`text`	`string`	the text of the message to show

.spec.podTemplate.options.podConfig.values[].spawner

Description

information for the spawner ui

Type

`object`

Required

`displayName`

Property	Type	Description
`description`	`string`	a description of the option
`displayName`	`string`	the display name of the option

Property	Type	Description
`hidden`	`boolean`	if this option should be hidden from the Workspace Spawner UI
`labels`	`array`	labels for the option

.spec.podTemplate.options.podConfig.values[].spawner.labels

Description

labels for the option

Type

`array`

.spec.podTemplate.options.podConfig.values[].spawner.labels[]

Type

`object`

Required

`key` `value`

Property	Type	Description
`key`	`string`	the key of the label

Property	Type	Description
`value`	`string`	the value of the label

.spec.podTemplate.options.podConfig.values[].spec

Description

the spec of the pod config

Type

`object`

Property	Type	Description
`affinity`	`object`	affinity configs for the pod
`nodeSelector`	`object`	node selector configs for the pod
`resources`	`object`	resource configs for the "main" container in the pod
`tolerations`	`array`	toleration configs for the pod

.spec.podTemplate.options.podConfig.values[].spec.affinity

Description

affinity configs for the pod

Type

`object`

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

.spec.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

Property	Type	Description
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	The scheduler will pr to schedule pods to nodes that satisfy the affinity expressions specified by this field, it may choose a node

Property	Type	Description
		violates one or more the expressions. The node that is most preferred is the one v the greatest sum of weights, i.e. for each node that meets all o scheduling requireme (resource request, requiredDuringSched affinity expressions, e 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.
`requiredDuringSchedulingIgnoredDuringExecution`	`object`	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 so point during pod execution (e.g. due to update), the system r or may not try to

Property	Type	Description
		eventually evict the p from its node.

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

Description

A node selector term, associated with the corresponding weight.

Type

`object`

Property	Type	Description
`matchExpressions`	`array`	A list of node selector requirements by node's labels.
`matchFields`	`array`	A list of node selector requirements by node's fields.

.spec.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.

Property	Type	Description
`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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE

xecution[].preference.matchFields

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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

Property	Type	Description
		element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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
`nodeSelectorTerms`	`array`	Required. A list of node selector terms. The terms are ORed.

.spec.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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
`matchExpressions`	`array`	A list of node selector requirements by node's labels.
`matchFields`	`array`	A list of node selector requirements by node's fields.

.spec.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

ecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.

Property	Type	Description
`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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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 pr to schedule pods to nodes that satisfy the affinity expressions specified by this field, it may choose a node violates one or more the expressions. The node that is most preferred is the one w the greatest sum of weights, i.e. for each node that meets all o scheduling requireme (resource request, requiredDuringSched affinity expressions, e compute a sum by iterating through the elements of this field adding "weight" to the sum if the node has p which matches the corresponding

Property	Type	Description
		podAffinityTerm; the node(s) with the high sum are the most preferred.
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	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 n try to eventually evict pod from its node. Wh there are multiple elements, the lists of nodes corresponding each podAffinityTerm intersected, i.e. all ter must be satisfied.

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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
`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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.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
`matchLabelKeys`	`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 `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).
`mismatchLabelKeys`	`array`	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).

Property	Type	Description
`namespaceSelector`	`object`	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.
`namespaces`	`array`	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".
`topologyKey`	`string`	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.podTemplate.options.podConfig.values[].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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringEx

ecution[].podAffinityTerm.mismatchLabelKeys[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringEx

ecution[].podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.

Property	Type	Description
`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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringEx

ecution[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExe

cution

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.podTemplate.options.podConfig.values[].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
`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
`matchLabelKeys`	`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 `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).
`mismatchLabelKeys`	`array`	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).

Property	Type	Description
`namespaceSelector`	`object`	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.
`namespaces`	`array`	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".
`topologyKey`	`string`	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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.
`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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExe

cution[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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 pr to schedule pods to nodes that satisfy the anti-affinity expressio specified by this field, it may choose a node

Property	Type	Description
		violates one or more the expressions. The node that is most preferred is the one v the greatest sum of weights, i.e. for each node that meets all o scheduling requireme (resource request, requiredDuringSched anti-affinity expressio etc.), compute a sum iterating through the elements of this field 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.
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	If the anti-affinity requirements specifie this field are not met scheduling time, the p will not be scheduled the node. If the anti-affinity requirements specified by this field cease to be met at so point during pod execution (e.g. due to

Property	Type	Description
		pod label update), the system may or may n 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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.
`matchLabelKeys`	`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 `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).
`mismatchLabelKeys`	`array`	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.

Property	Type	Description
		Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
`namespaceSelector`	`object`	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.
`namespaces`	`array`	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".
`topologyKey`	`string`	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.podTemplate.options.podConfig.values[].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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
`matchLabels`	`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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDurin

gExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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
`labelSelector`	`object`	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
`matchLabelKeys`	`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 `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).
`mismatchLabelKeys`	`array`	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

Property	Type	Description
		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).
`namespaceSelector`	`object`	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.
`namespaces`	`array`	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
`topologyKey`	`string`	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.podTemplate.options.podConfig.values[].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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].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
`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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuring

Execution[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].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
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.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.podTemplate.options.podConfig.values[].spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.podTemplate.options.podConfig.values[].spec.nodeSelector

Description

node selector configs for the pod

Type

`object`

.spec.podTemplate.options.podConfig.values[].spec.resources

Description

resource configs for the "main" container in the pod

Type

`object`

Property	Type	Description
`claims`	`array`	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.
`limits`	`object`	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
`requests`	`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.podTemplate.options.podConfig.values[].spec.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.podTemplate.options.podConfig.values[].spec.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
`request`	`string`	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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].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.podTemplate.options.podConfig.values[].spec.tolerations

Description

toleration configs for the pod

Type

`array`

.spec.podTemplate.options.podConfig.values[].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
`effect`	`string`	Effect indicates the taint effect to match. Empty means match all taint effects. When specified,

Property	Type	Description
		allowed values are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	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.
`operator`	`string`	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.
`tolerationSeconds`	`integer`	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.
`value`	`string`	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.podTemplate.podMetadata

Description

metadata for Workspace Pods (MUTABLE)

Type

`object`

Property	Type	Description
`annotations`	`object`	annotations to be applied to the Pod resource
`labels`	`object`	labels to be applied to the Pod resource

.spec.podTemplate.podMetadata.annotations

Description

annotations to be applied to the Pod resource

Type

`object`

.spec.podTemplate.podMetadata.labels

Description

labels to be applied to the Pod resource

Type

`object`

.spec.podTemplate.probes

Description

standard probes to determine Container health (MUTABLE)

Type

`object`

Property	Type	Description
`livenessProbe`	`object`	the liveness probe for the main container
`readinessProbe`	`object`	the readiness probe for the main container
`startupProbe`	`object`	the startup probe for the main container

.spec.podTemplate.probes.livenessProbe

Description

the liveness probe for the main container

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. Minimum value is 1.
`grpc`	`object`	GRPC specifies an action involving a GRP

Property	Type	Description
`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 prob 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 ru pod are sent a termination signal and the ti processes are forcibly halted with a kill sig 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

Property	Type	Description
		(no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is us
`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.podTemplate.probes.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.podTemplate.probes.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.podTemplate.probes.livenessProbe.exec.command[]

Type

`string`

.spec.podTemplate.probes.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

Property	Type	Description
		https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.podTemplate.probes.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.

Property	Type	Description
`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.podTemplate.probes.livenessProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.podTemplate.probes.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.podTemplate.probes.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.podTemplate.probes.readinessProbe

Description

the readiness probe for the main container

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. 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 prob 10 seconds. Minimum value is 1.

Property	Type	Description
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. D 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 rur pod are sent a termination signal and the ti processes are forcibly halted with a kill sigr 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 ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is us
`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.podTemplate.probes.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.podTemplate.probes.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.podTemplate.probes.readinessProbe.exec.command[]

Type

`string`

.spec.podTemplate.probes.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.podTemplate.probes.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.podTemplate.probes.readinessProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.podTemplate.probes.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.podTemplate.probes.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.
`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.podTemplate.probes.startupProbe

Description

the startup probe for the main container

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. Minimum value is 1.
`grpc`	`object`	GRPC specifies an action involving a GRP
`httpGet`	`object`	HTTPGet specifies the http request to perf

Property	Type	Description
`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 prob 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

Property	Type	Description
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes ru pod are sent a termination signal and the t processes are forcibly halted with a kill sig 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 ProbeTerminationGraceP 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.podTemplate.probes.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 explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.podTemplate.probes.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.podTemplate.probes.startupProbe.exec.command[]

Type

`string`

.spec.podTemplate.probes.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.podTemplate.probes.startupProbe.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
`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.podTemplate.probes.startupProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.podTemplate.probes.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.podTemplate.probes.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.
`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.podTemplate.securityContext

Description

security context for Workspace Pods (MUTABLE)

Type

`object`

Property	Type	Description
`appArmorProfile`	`object`	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.
`fsGroup`	`integer`	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 spec.os.name is windows.
`fsGroupChangePolicy`	`string`	fsGroupChangePolicy defines behavior of changing ownership and permission of the

Property	Type	Description
		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 "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.

Property	Type	Description
`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 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

Property	Type	Description
		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.
`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

Property	Type	Description
		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.podTemplate.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
`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".

Property	Type	Description
`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.podTemplate.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.podTemplate.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
`localhostProfile`	`string`	localhostProfile 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.
`type`	`string`	type 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.podTemplate.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.podTemplate.securityContext.supplementalGroups[]

Type

`integer`

.spec.podTemplate.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.podTemplate.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.podTemplate.securityContext.windowsOptions

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
`gmsaCredentialSpec`	`string`	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 GMSACredentialSpecName field.
`gmsaCredentialSpecName`	`string`	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
`hostProcess`	`boolean`	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.
`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.podTemplate.serviceAccount

Description

service account configs for Workspace Pods

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	the name of the ServiceAccount (NOT MUTABLE) • this Service Account MUST already exist in the Namespace of the Workspace, the controller will NOT create it • we will not show this WorkspaceKind in the Spawner UI if the SA does not exist in the Namespace

.spec.podTemplate.volumeMounts

Description

volume mount paths

Type

`object`

Required

`home`

Property	Type	Description
`home`	`string`	the path to mount the home PVC (NOT MUTABLE)

.spec.spawner

Description

spawner config determines how the WorkspaceKind is displayed in the Workspace Spawner UI

Type

`object`

Required

`description` `displayName` `icon` `logo`

Property	Type	Description
`deprecated`	`boolean`	if this WorkspaceKind is deprecated
`deprecationMessage`	`string`	a message to show in Workspace Spawner UI when the WorkspaceKind is deprecated
`description`	`string`	the description of the WorkspaceKind
`displayName`	`string`	the display name of the WorkspaceKind
`hidden`	`boolean`	if this WorkspaceKind should be hidden from the Workspace Spawner UI
`icon`	`object`	the icon of the WorkspaceKind • a small (favicon-sized) icon used in the Workspace Spawner UI
`logo`	`object`	the logo of the WorkspaceKind • a 1:1 (card size) logo used in the Workspace Spawner UI

.spec.spawner.icon

Description

the icon of the WorkspaceKind - a small (favicon-sized) icon used in the Workspace Spawner UI

Type

`object`

Property	Type	Description
`configMap`	`object`	
`url`	`string`	

.spec.spawner.icon.configMap

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	
`name`	`string`	

.spec.spawner.logo

Description

the logo of the WorkspaceKind - a 1:1 (card size) logo used in the Workspace Spawner UI

Type

`object`

Property	Type	Description
`configMap`	`object`	
`url`	`string`	

.spec.spawner.logo.configMap

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	
`name`	`string`	

.status

Description

WorkspaceKindStatus defines the observed state of WorkspaceKind

Type

`object`

Required

`podTemplateOptions` `workspaces`

Property	Type	Description
`podTemplateOptions`	`object`	metrics for podTemplate options

Property	Type	Description
`workspaces`	`integer`	the number of Workspaces that are using this WorkspaceKind

.status.podTemplateOptions

Description

metrics for podTemplate options

Type

`object`

Required

`imageConfig` `podConfig`

Property	Type	Description
`imageConfig`	`array`	metrics about the imageConfig options
`podConfig`	`array`	metrics about the podConfig options

.status.podTemplateOptions.imageConfig

Description

metrics about the imageConfig options

Type

`array`

.status.podTemplateOptions.imageConfig[]

Type

`object`

Required

`id` `workspaces`

Property	Type	Description
`id`	`string`	the id of the option
`workspaces`	`integer`	the number of Workspaces currently using the option

.status.podTemplateOptions.podConfig

Description

metrics about the podConfig options

Type

`array`

.status.podTemplateOptions.podConfig[]

Type

`object`

Required

`id` `workspaces`

Property	Type	Description
`id`	`string`	the id of the option

Property	Type	Description
`workspaces`	`integer`	the number of Workspaces currently using the option

API Endpoints

The following API endpoints are available:

- `/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspacekinds`
  - `DELETE`: delete collection of WorkspaceKind
  - `GET`: list objects of kind WorkspaceKind
  - `POST`: create a new WorkspaceKind
- `/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspacekinds/{name}`
  - `DELETE`: delete the specified WorkspaceKind
  - `GET`: read the specified WorkspaceKind
  - `PATCH`: partially update the specified WorkspaceKind
  - `PUT`: replace the specified WorkspaceKind
- `/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspacekinds/{name}/status`
  - `GET`: read status of the specified WorkspaceKind
  - `PATCH`: partially update status of the specified WorkspaceKind
  - `PUT`: replace status of the specified WorkspaceKind

/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspacekinds

HTTP method

`DELETE`

Description

delete collection of WorkspaceKind

HTTP responses

HTTP code	Response body
200 - OK	`Status↗` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind WorkspaceKind

HTTP responses

HTTP code	Response body
200 - OK	`WorkspaceKindList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new WorkspaceKind

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
`fieldValidation`	`string`	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
`body`	`WorkspaceKind` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`WorkspaceKind` schema
201 - Created	`WorkspaceKind` schema
202 - Accepted	`WorkspaceKind` schema
401 - Unauthorized	Empty

/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspacekinds/{name}

HTTP method

`DELETE`

Description

delete the specified WorkspaceKind

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 WorkspaceKind

HTTP responses

HTTP code	Response body
200 - OK	`WorkspaceKind` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified WorkspaceKind

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
`fieldValidation`	`string`	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	`WorkspaceKind` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified WorkspaceKind

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

Parameter	Type	Description
`fieldValidation`	`string`	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
`body`	`WorkspaceKind` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`WorkspaceKind` schema
201 - Created	`WorkspaceKind` schema
401 - Unauthorized	Empty

/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspacekinds/{name}/status

HTTP method

`GET`

Description

read status of the specified WorkspaceKind

HTTP responses

HTTP code	Response body
200 - OK	`WorkspaceKind` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified WorkspaceKind

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
`fieldValidation`	`string`	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	`WorkspaceKind` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified WorkspaceKind

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
`fieldValidation`	`string`	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`	`WorkspaceKind` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`WorkspaceKind` schema
201 - Created	`WorkspaceKind` schema
401 - Unauthorized	Empty

Workspace [kubeflow.org/v1beta1]

Description

Workspace is the Schema for the Workspaces API

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	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 ↗

Property	Type	Description
`kind`	`string`	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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta` ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	WorkspaceSpec defines the desired state of Workspace
`status`	`object`	WorkspaceStatus defines the observed state of Workspace

.spec

Description

WorkspaceSpec defines the desired state of Workspace

Type

`object`

Required

`kind` `podTemplate`

Property	Type	Description
`deferUpdates`	`boolean`	if true, pending updates are NOT applied when the Workspace is paused if false, pending updates are applied when the Workspace is paused
`kind`	`string`	the WorkspaceKind to use
`paused`	`boolean`	if the workspace is paused (no pods running)
`podTemplate`	`object`	options for "podTemplate"-type WorkspaceKinds

.spec.podTemplate

Description

options for "podTemplate"-type WorkspaceKinds

Type

`object`

Required

`options` `volumes`

Property	Type	Description
`options`	`object`	the selected podTemplate options
`podMetadata`	`object`	metadata to be applied to the Pod resource

Property	Type	Description
`volumes`	`object`	volume configs

.spec.podTemplate.options

Description

the selected podTemplate options

Type

`object`

Required

`imageConfig` `podConfig`

Property	Type	Description
`imageConfig`	`string`	the id of an imageConfig option • options are defined in WorkspaceKind under `spec.podTemplate.options.imageConfig.values[]`
`podConfig`	`string`	the id of a podConfig option • options are defined in WorkspaceKind under `spec.podTemplate.options.podConfig.values[]`

.spec.podTemplate.podMetadata

Description

metadata to be applied to the Pod resource

Type

`object`

Property	Type	Description
`annotations`	`object`	annotations to be applied to the Pod resource
`labels`	`object`	labels to be applied to the Pod resource

.spec.podTemplate.podMetadata.annotations

Description

annotations to be applied to the Pod resource

Type

`object`

.spec.podTemplate.podMetadata.labels

Description

labels to be applied to the Pod resource

Type

`object`

.spec.podTemplate.volumes

Description

volume configs

Type

`object`

Property	Type	Description
`data`	`array`	additional PVCs to mount • these PVC must already exist in the Namespace • the same PVC can be mounted multiple times with different `mountPaths` • if `readOnly` is false, the PVC must be RWX (ReadWriteMany, ReadWriteOnce) • if `readOnly` is true, the PVC must be ReadOnlyMany
`home`	`string`	the name of the PVC to mount as the home volume • this PVC must already exist in the Namespace • this PVC must be RWX (ReadWriteMany, ReadWriteOnce) • the mount path is defined in the WorkspaceKind under `spec.podTemplate.volumeMounts.home`

.spec.podTemplate.volumes.data

Description

additional PVCs to mount - these PVC must already exist in the Namespace - the same PVC can be mounted multiple times with different `mountPaths` - if `readOnly` is false, the PVC must be RWX (ReadWriteMany, ReadWriteOnce) - if `readOnly` is true, the PVC must be ReadOnlyMany

Type

`array`

.spec.podTemplate.volumes.data[]

Type

`object`

Required

`mountPath` `pvcName`

Property	Type	Description
`mountPath`	`string`	the mount path for the PVC
`pvcName`	`string`	the name of the PVC to mount
`readOnly`	`boolean`	if the PVC should be mounted as ReadOnly

.status

Description

WorkspaceStatus defines the observed state of Workspace

Type

`object`

Required

`activity` `pauseTime` `pendingRestart` `podTemplateOptions` `podTemplatePod` `state` `stateMessage`

Property	Type	Description
`activity`	`object`	activity information for the Workspace, used to determine when to cull

Property	Type	Description
`pauseTime`	`integer`	the time when the Workspace was paused (UNIX epoch) • set to 0 when the Workspace is NOT paused
`pendingRestart`	`boolean`	if the current Pod does not reflect the current "desired" state • true if any `spec.podTemplate.options` have a redirect and so will be patched on the next restart • true if the WorkspaceKind has changed one of its common `podTemplate` fields like `podMetadata`, `probes`, `extraEnv`, or `containerSecurityContext`
`podTemplateOptions`	`object`	information about the current podTemplate options (only set for WorkspaceKind of podTemplate kind)
`podTemplatePod`	`object`	information about the Pod managed by this Workspace (only set for WorkspaceKind of podTemplate kind)
`state`	`string`	the current state of the Workspace
`stateMessage`	`string`	a human-readable message about the state of the Workspace

Property	Type	Description
		• WARNING: this field is NOT FOR MACHINE USE, subject to change without notice

.status.activity

Description

activity information for the Workspace, used to determine when to cull

Type

`object`

Required

`lastActivity` `lastUpdate`

Property	Type	Description
`lastActivity`	`integer`	the last time activity was observed on the Workspace (UNIX epoch)
`lastUpdate`	`integer`	the last time we checked for activity on the Workspace (UNIX epoch)

.status.podTemplateOptions

Description

information about the current podTemplate options (only set for WorkspaceKind of podTemplate kind)

Type

`object`

Required

`imageConfig` `podConfig`

Property	Type	Description
`imageConfig`	`object`	info about the current imageConfig option
`podConfig`	`object`	info about the current podConfig option

.status.podTemplateOptions.imageConfig

Description

info about the current imageConfig option

Type

`object`

Property	Type	Description
`desired`	`string`	the option id which will take effect after the next restart
`redirectChain`	`array`	the chain from the current option to the desired option

.status.podTemplateOptions.imageConfig.redirectChain

Description

the chain from the current option to the desired option

Type

`array`

.status.podTemplateOptions.imageConfig.redirectChain[]

Type

`object`

Required

`source` `target`

Property	Type	Description
`source`	`string`	the source option id
`target`	`string`	the target option id

.status.podTemplateOptions.podConfig

Description

info about the current podConfig option

Type

`object`

Property	Type	Description
`desired`	`string`	the option id which will take effect after the next restart
`redirectChain`	`array`	the chain from the current option to the desired option

.status.podTemplateOptions.podConfig.redirectChain

Description

the chain from the current option to the desired option

Type

`array`

.status.podTemplateOptions.podConfig.redirectChain[]

Type

`object`

Required

`source` `target`

Property	Type	Description
`source`	`string`	the source option id
`target`	`string`	the target option id

.status.podTemplatePod

Description

information about the Pod managed by this Workspace (only set for WorkspaceKind of podTemplate kind)

Type

`object`

Required

`name`

Property	Type	Description
`containers`	`array`	information about the Pod's containers
`initContainers`	`array`	information about the Pod's initContainers
`name`	`string`	the name of the Pod resource

.status.podTemplatePod.containers

Description

information about the Pod's containers

Type

`array`

.status.podTemplatePod.containers[]

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	the name of the container

.status.podTemplatePod.initContainers

Description

information about the Pod's initContainers

Type

`array`

.status.podTemplatePod.initContainers[]

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	the name of the container

API Endpoints

The following API endpoints are available:

- `/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspaces`
  - `DELETE`: delete collection of Workspace
  - `GET`: list objects of kind Workspace
  - `POST`: create a new Workspace
- `/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspaces/{name}`
  - `DELETE`: delete the specified Workspace
  - `GET`: read the specified Workspace
  - `PATCH`: partially update the specified Workspace
  - `PUT`: replace the specified Workspace

- `/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspaces/{name}/status`
  - `GET`: read status of the specified Workspace
  - `PATCH`: partially update status of the specified Workspace
  - `PUT`: replace status of the specified Workspace

/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspaces

HTTP method

`DELETE`

Description

delete collection of Workspace

HTTP responses

HTTP code	Response body
200 - OK	`Status↗` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind Workspace

HTTP responses

HTTP code	Response body
200 - OK	`WorkspaceList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new Workspace

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
`fieldValidation`	`string`	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
`body`	`Workspace` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Workspace` schema
201 - Created	`Workspace` schema
202 - Accepted	`Workspace` schema
401 - Unauthorized	Empty

/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspaces/{name}

HTTP method

`DELETE`

Description

delete the specified Workspace

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 Workspace

HTTP responses

HTTP code	Response body
200 - OK	`Workspace` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Workspace

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
`fieldValidation`	`string`	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	`Workspace` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Workspace

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
`fieldValidation`	`string`	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`	`Workspace` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Workspace` schema
201 - Created	`Workspace` schema
401 - Unauthorized	Empty

/apis/kubeflow.org/v1beta1/namespaces/{namespace}/workspaces/{name}/status

HTTP method

`GET`

Description

read status of the specified Workspace

HTTP responses

HTTP code	Response body
200 - OK	`Workspace` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified Workspace

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
`fieldValidation`	`string`	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	`Workspace` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Workspace

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

Parameter	Type	Description
`fieldValidation`	`string`	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
`body`	`Workspace` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Workspace` schema
201 - Created	`Workspace` schema
401 - Unauthorized	Empty