Operator APIs

CatalogSource [operators.core(

ClusterServiceVersion [operato

InstallPlan [

OLMConfig [operators.coreos.c

OperatorCondition [operators.c

OperatorGr

Subscription [operators.coreos

Operator [operators.operators.coreos.com/v1

CatalogSource [operators.coreos.com/v1alpha1]

Description

CatalogSource is a repository of CSVs, CRDs, and operator packages.

Type

`object`

Required

`metadata` `spec`

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`	
`status`	`object`	

.spec

Type

`object`

Required

`sourceType`

Property	Type	Description
`address`	`string`	Address is a host that OLM can use to connect to a pre-existing registry. Format: : Only used when SourceType = SourceTypeGrpc. Ignored when the Image field is set.

Property	Type	Description
`configMap`	`string`	ConfigMap is the name of the ConfigMap to be used to back a configmap-server registry. Only used when SourceType = SourceTypeConfigmap or SourceTypeInternal.
`description`	`string`	
`displayName`	`string`	Metadata
`grpcPodConfig`	`object`	GrpcPodConfig exposes different overrides for the pod spec of the CatalogSource Pod. Only used when SourceType = SourceTypeGrpc and Image is set.
`icon`	`object`	
`image`	`string`	Image is an operator-registry container image to instantiate a registry-server with. Only used when SourceType = SourceTypeGrpc. If present, the address field is ignored.
`priority`	`integer`	Priority field assigns a weight to the catalog source to prioritize them so that it can be consumed by the dependency resolver. Usage: Higher weight indicates that this catalog source is preferred over lower weighted catalog sources during dependency resolution. The range of the priority value can go from positive to negative in the range of int32. The default value to a catalog source with unassigned priority would be 0. The catalog source with the same priority

Property	Type	Description
		values will be ranked lexicographically based on its name.
`publisher`	`string`	
`secrets`	`array`	Secrets represent set of secrets that can be used to access the contents of the catalog. It is best to keep this list small, since each will need to be tried for every catalog entry.
`sourceType`	`string`	SourceType is the type of source
`updateStrategy`	`object`	UpdateStrategy defines how updated catalog source images can be discovered Consists of an interval that defines polling duration and an embedded strategy type

.spec.grpcPodConfig

Description

GrpcPodConfig exposes different overrides for the pod spec of the CatalogSource Pod. Only used when SourceType = SourceTypeGrpc and Image is set.

Type

`object`

Property	Type	Description
`nodeSelector`	`object`	NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match

Property	Type	Description
		a node's labels for the pod to be scheduled on that node.
`priorityClassName`	`string`	If specified, indicates the pod's priority. If not specified, the pod priority will be default or zero if there is no default.
`tolerations`	`array`	Tolerations are the catalog source's pod's tolerations.

.spec.grpcPodConfig.nodeSelector

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node.

Type

`object`

.spec.grpcPodConfig.tolerations

Description

Tolerations are the catalog source's pod's tolerations.

Type

`array`

.spec.grpcPodConfig.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, 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.

Property	Type	Description
`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.icon

Type

`object`

Required

`base64data` `mediatype`

Property	Type	Description
`base64data`	`string`	
`mediatype`	`string`	

.spec.secrets

Description

Secrets represent set of secrets that can be used to access the contents of the catalog. It is best to keep this list small, since each will need to be tried for every catalog entry.

Type

`array`

.spec.secrets[]

Type

`string`

.spec.updateStrategy

Description

UpdateStrategy defines how updated catalog source images can be discovered Consists of an interval that defines polling duration and an embedded strategy type

Type

`object`

Property	Type	Description
`registryPoll`	`object`	

.spec.updateStrategy.registryPoll

Type

`object`

Property	Type	Description
`interval`	`string`	Interval is used to determine the time interval between checks of the latest catalog source version. The catalog operator polls to see if a new version of the catalog source is available. If available, the latest image is pulled and gRPC traffic is directed to the latest catalog source.

.status

Type

`object`

Property	Type	Description
`conditions`	`array`	Represents the state of a CatalogSource. Note that Message and Reason represent the original status information, which may be migrated to be conditions based in the future. Any new features introduced will use conditions.
`configMapReference`	`object`	
`connectionState`	`object`	
`latestImageRegistryPoll`	`string`	The last time the CatalogSource image registry has been polled to ensure the image is up-to-date
`message`	`string`	A human readable message indicating details about why the CatalogSource is in this condition.
`reason`	`string`	Reason is the reason the CatalogSource was transitioned to its current state.
`registryService`	`object`	

.status.conditions

Description

Represents the state of a CatalogSource. Note that Message and Reason represent the original status information, which may be migrated to be conditions based in the future. Any

new features introduced will use conditions.

Type

`array`

.status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"` // other fields }

Type

`object`

Required

`lastTransitionTime` `message` `reason` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.

Property	Type	Description
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. --- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

.status.configMapReference

Type

`object`

Required

`name` `namespace`

Property	Type	Description
`lastUpdateTime`	`string`	
`name`	`string`	
`namespace`	`string`	
`resourceVersion`	`string`	
`uid`	`string`	UID is a type that holds unique ID values, including UUIDs. Because we don't ONLY use UUIDs, this is an alias to string. Being a type captures intent and helps make sure that UIDs and names do not get conflated.

.status.connectionState

Type

`object`

Required

`lastObservedState`

Property	Type	Description
`address`	`string`	
`lastConnect`	`string`	

Property	Type	Description
`lastObservedState`	`string`	

.status.registryService

Type

`object`

Property	Type	Description
`createdAt`	`string`	
`port`	`string`	
`protocol`	`string`	
`serviceName`	`string`	
`serviceNamespace`	`string`	

API Endpoints

The following API endpoints are available:

- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsources`
  - `DELETE`: delete collection of CatalogSource
  - `GET`: list objects of kind CatalogSource
  - `POST`: create a new CatalogSource
- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsources/{name}`
  - `DELETE`: delete the specified CatalogSource
  - `GET`: read the specified CatalogSource

- `PATCH`: partially update the specified CatalogSource
- `PUT`: replace the specified CatalogSource

- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsources/{name}/status`
  - `GET`: read status of the specified CatalogSource
  - `PATCH`: partially update status of the specified CatalogSource
  - `PUT`: replace status of the specified CatalogSource

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsources

HTTP method

`DELETE`

Description

delete collection of CatalogSource

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind CatalogSource

HTTP responses

HTTP code	Response body
200 - OK	`CatalogSourceList` schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new CatalogSource

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`	`CatalogSource` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsources/{name}

HTTP method

`DELETE`

Description

delete the specified CatalogSource

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 CatalogSource

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified CatalogSource

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:

Parameter	Type	Description
		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	`CatalogSource` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified CatalogSource

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`	`CatalogSource` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsources/{name}/status

HTTP method

`GET`

Description

read status of the specified CatalogSource

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified CatalogSource

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	`CatalogSource` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified CatalogSource

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`	`CatalogSource` schema	`application/json` formatted

HTTP responses

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

ClusterServiceVersion [operators.coreos.com/v1alpha1]

Description

ClusterServiceVersion is a Custom Resource of type `ClusterServiceVersionSpec`.

Type

`object`

Required

`metadata` `spec`

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`	ClusterServiceVersionSpec declarations tell OLM how to install an operator that can manage apps for a given version.
`status`	`object`	ClusterServiceVersionStatus represents information about the status of a CSV. Status may trail the actual state of a system.

.spec

Description

ClusterServiceVersionSpec declarations tell OLM how to install an operator that can manage apps for a given version.

Type

`object`

Required

`displayName` `install`

Property	Type	Description
`annotations`	`object`	Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata.
`apiservicedefinitions`	`object`	APIServiceDefinitions declares all of the extension apis managed or required by an operator being ran by ClusterServiceVersion.
`cleanup`	`object`	Cleanup specifies the cleanup behaviour when the CSV gets deleted
`customresourcedefinitions`	`object`	CustomResourceDefinitions declares all of the CRDs managed or required by an operator being ran by ClusterServiceVersion. If the CRD is present in the Owned list, it is implicitly required.
`description`	`string`	
`displayName`	`string`	
`icon`	`array`	

Property	Type	Description
`install`	`object`	NamedInstallStrategy represents the block of an ClusterServiceVersion resource where the install strategy is specified.
`installModes`	`array`	InstallModes specify supported installation types
`keywords`	`array`	
`labels`	`object`	Map of string keys and values that can be used to organize and categorize (scope and select) objects.
`links`	`array`	
`maintainers`	`array`	
`maturity`	`string`	
`minKubeVersion`	`string`	
`nativeAPIs`	`array`	
`provider`	`object`	
`relatedImages`	`array`	List any related images, or other container images that your Operator might require to perform their functions. This list should also include operand images as well. All image references should be specified by digest (SHA) and not by tag. This field is

Property	Type	Description
		only used during catalog creation and plays no part in cluster runtime.
`replaces`	`string`	The name of a CSV this one replaces. Should match the `metadata.Name` field of the old CSV.
`selector`	`object`	Label selector for related resources.
`skips`	`array`	The name(s) of one or more CSV(s) that should be skipped in the upgrade graph. Should match the `metadata.Name` field of the CSV that should be skipped. This field is only used during catalog creation and plays no part in cluster runtime.
`version`	`string`	OperatorVersion is a wrapper around semver.Version which supports correct marshaling to YAML and JSON.
`webhookdefinitions`	`array`	

.spec.annotations

Description

Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata.

Type

`object`

.spec.apiservicedefinitions

Description

APIServiceDefinitions declares all of the extension apis managed or required by an operator being ran by ClusterServiceVersion.

Type

`object`

Property	Type	Description
`owned`	`array`	
`required`	`array`	

.spec.apiservicedefinitions.owned

Type

`array`

.spec.apiservicedefinitions.owned[]

Description

APIServiceDescription provides details to OLM about apis provided via aggregation

Type

`object`

Required

`group` `kind` `name` `version`

Property	Type	Description
`actionDescriptors`	`array`	

Property	Type	Description
`containerPort`	`integer`	
`deploymentName`	`string`	
`description`	`string`	
`displayName`	`string`	
`group`	`string`	
`kind`	`string`	
`name`	`string`	
`resources`	`array`	
`specDescriptors`	`array`	
`statusDescriptors`	`array`	
`version`	`string`	

.spec.apiservicedefinitions.owned[].actionDescriptors

Type

`array`

.spec.apiservicedefinitions.owned[].actionDescriptors[]

Description

ActionDescriptor describes a declarative action that can be performed on a custom resource instance

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.apiservicedefinitions.owned[].actionDescriptors[].x-descriptors

Type

`array`

.spec.apiservicedefinitions.owned[].actionDescriptors[].x-descriptors[]

Type

`string`

.spec.apiservicedefinitions.owned[].resources

Type

`array`

.spec.apiservicedefinitions.owned[].resources[]

Description

APIResourceReference is a Kubernetes resource type used by a custom resource

Type

`object`

Required

`kind` `name` `version`

Property	Type	Description
`kind`	`string`	
`name`	`string`	
`version`	`string`	

.spec.apiservicedefinitions.owned[].specDescriptors

Type

`array`

.spec.apiservicedefinitions.owned[].specDescriptors[]

Description

SpecDescriptor describes a field in a spec block of a CRD so that OLM can consume it

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.apiservicedefinitions.owned[].specDescriptors[].x-descriptors

Type

`array`

.spec.apiservicedefinitions.owned[].specDescriptors[].x-descriptors[]

Type

`string`

.spec.apiservicedefinitions.owned[].statusDescriptors

Type

`array`

.spec.apiservicedefinitions.owned[].statusDescriptors[]

Description

StatusDescriptor describes a field in a status block of a CRD so that OLM can consume it

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.apiservicedefinitions.owned[].statusDescriptors[].x-descriptors

Type

`array`

.spec.apiservicedefinitions.owned[].statusDescriptors[].x-descriptors[]

Type

`string`

.spec.apiservicedefinitions.required

Type

`array`

.spec.apiservicedefinitions.required[]

Description

APIServiceDescription provides details to OLM about apis provided via aggregation

Type

`object`

Required

`group` `kind` `name` `version`

Property	Type	Description
`actionDescriptors`	`array`	
`containerPort`	`integer`	
`deploymentName`	`string`	
`description`	`string`	
`displayName`	`string`	
`group`	`string`	
`kind`	`string`	
`name`	`string`	
`resources`	`array`	

Property	Type	Description
`specDescriptors`	`array`	
`statusDescriptors`	`array`	
`version`	`string`	

.spec.apiservicedefinitions.required[].actionDescriptors

Type

`array`

.spec.apiservicedefinitions.required[].actionDescriptors[]

Description

ActionDescriptor describes a declarative action that can be performed on a custom resource instance

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	

Property	Type	Description
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.apiservicedefinitions.required[].actionDescriptors[].x-descriptors

Type

`array`

.spec.apiservicedefinitions.required[].actionDescriptors[].x-descriptors[]

Type

`string`

.spec.apiservicedefinitions.required[].resources

Type

`array`

.spec.apiservicedefinitions.required[].resources[]

Description

APIResourceReference is a Kubernetes resource type used by a custom resource

Type

`object`

Required

`kind` `name` `version`

Property	Type	Description
`kind`	`string`	
`name`	`string`	
`version`	`string`	

.spec.apiservicedefinitions.required[].specDescriptors

Type

`array`

.spec.apiservicedefinitions.required[].specDescriptors[]

Description

SpecDescriptor describes a field in a spec block of a CRD so that OLM can consume it

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	

Property	Type	Description
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.apiservicedefinitions.required[].specDescriptors[].x-descriptors

Type

`array`

.spec.apiservicedefinitions.required[].specDescriptors[].x-descriptors[]

Type

`string`

.spec.apiservicedefinitions.required[].statusDescriptors

Type

`array`

.spec.apiservicedefinitions.required[].statusDescriptors[]

Description

StatusDescriptor describes a field in a status block of a CRD so that OLM can consume it

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.apiservicedefinitions.required[].statusDescriptors[].x-descriptors

Type

`array`

.spec.apiservicedefinitions.required[].statusDescriptors[].x-descriptors[]

Type

`string`

.spec.cleanup

Description

Cleanup specifies the cleanup behaviour when the CSV gets deleted

Type

`object`

Required

`enabled`

Property	Type	Description
`enabled`	`boolean`	

.spec.customresourcedefinitions

Description

CustomResourceDefinitions declares all of the CRDs managed or required by an operator being ran by ClusterServiceVersion. If the CRD is present in the Owned list, it is implicitly required.

Type

`object`

Property	Type	Description
`owned`	`array`	
`required`	`array`	

.spec.customresourcedefinitions.owned

Type

`array`

.spec.customresourcedefinitions.owned[]

Description

CRDDescription provides details to OLM about the CRDs

Type

`object`

Required

`kind` `name` `version`

Property	Type	Description
`actionDescriptors`	`array`	
`description`	`string`	
`displayName`	`string`	
`kind`	`string`	
`name`	`string`	
`resources`	`array`	
`specDescriptors`	`array`	
`statusDescriptors`	`array`	
`version`	`string`	

.spec.customresourcedefinitions.owned[].actionDescriptors

Type

`array`

.spec.customresourcedefinitions.owned[].actionDescriptors[]

Description

ActionDescriptor describes a declarative action that can be performed on a custom resource instance

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.customresourcedefinitions.owned[].actionDescriptors[].x-descriptors

Type

`array`

.spec.customresourcedefinitions.owned[].actionDescriptors[].x-descriptors[]

Type

`string`

.spec.customresourcedefinitions.owned[].resources

Type

`array`

.spec.customresourcedefinitions.owned[].resources[]

Description

APIResourceReference is a Kubernetes resource type used by a custom resource

Type

`object`

Required

`kind` `name` `version`

Property	Type	Description
`kind`	`string`	
`name`	`string`	
`version`	`string`	

.spec.customresourcedefinitions.owned[].specDescriptors

Type

`array`

.spec.customresourcedefinitions.owned[].specDescriptors[]

Description

SpecDescriptor describes a field in a spec block of a CRD so that OLM can consume it

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.customresourcedefinitions.owned[].specDescriptors[].x-descriptors

Type

`array`

.spec.customresourcedefinitions.owned[].specDescriptors[].x-descriptors[]

Type

`string`

.spec.customresourcedefinitions.owned[].statusDescriptors

Type

`array`

.spec.customresourcedefinitions.owned[].statusDescriptors[]

Description

StatusDescriptor describes a field in a status block of a CRD so that OLM can consume it

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.customresourcedefinitions.owned[].statusDescriptors[].x-descriptors

Type

`array`

.spec.customresourcedefinitions.owned[].statusDescriptors[].x-descriptors[]

Type

`string`

.spec.customresourcedefinitions.required

Type

`array`

.spec.customresourcedefinitions.required[]

Description

CRDDescription provides details to OLM about the CRDs

Type

`object`

Required

`kind` `name` `version`

Property	Type	Description
`actionDescriptors`	`array`	
`description`	`string`	

Property	Type	Description
`displayName`	`string`	
`kind`	`string`	
`name`	`string`	
`resources`	`array`	
`specDescriptors`	`array`	
`statusDescriptors`	`array`	
`version`	`string`	

.spec.customresourcedefinitions.required[].actionDescriptors

Type

`array`

.spec.customresourcedefinitions.required[].actionDescriptors[]

Description

ActionDescriptor describes a declarative action that can be performed on a custom resource instance

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.customresourcedefinitions.required[].actionDescriptors[].x-descriptors

Type

`array`

.spec.customresourcedefinitions.required[].actionDescriptors[].x-descriptors[]

Type

`string`

.spec.customresourcedefinitions.required[].resources

Type

`array`

.spec.customresourcedefinitions.required[].resources[]

Description

APIResourceReference is a Kubernetes resource type used by a custom resource

Type

`object`

Required

`kind` `name` `version`

Property	Type	Description
`kind`	`string`	
`name`	`string`	
`version`	`string`	

.spec.customresourcedefinitions.required[].specDescriptors

Type

`array`

.spec.customresourcedefinitions.required[].specDescriptors[]

Description

SpecDescriptor describes a field in a spec block of a CRD so that OLM can consume it

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.customresourcedefinitions.required[].specDescriptors[].x-descriptors

Type

`array`

.spec.customresourcedefinitions.required[].specDescriptors[].x-descriptors[]

Type

`string`

.spec.customresourcedefinitions.required[].statusDescriptors

Type

`array`

.spec.customresourcedefinitions.required[].statusDescriptors[]

Description

StatusDescriptor describes a field in a status block of a CRD so that OLM can consume it

Type

`object`

Required

`path`

Property	Type	Description
`description`	`string`	
`displayName`	`string`	
`path`	`string`	
`value`	`string`	RawMessage is a raw encoded JSON value. It implements Marshaler and Unmarshaler and can be used to delay JSON decoding or precompute a JSON encoding.
`x-descriptors`	`array`	

.spec.customresourcedefinitions.required[].statusDescriptors[].x-descriptors

Type

`array`

.spec.customresourcedefinitions.required[].statusDescriptors[].x-descriptors[]

Type

`string`

.spec.icon

Type

`array`

.spec.icon[]

Type

`object`

Required

`base64data` `mediatype`

Property	Type	Description
`base64data`	`string`	
`mediatype`	`string`	

.spec.install

Description

NamedInstallStrategy represents the block of an ClusterServiceVersion resource where the install strategy is specified.

Type

`object`

Required

`strategy`

Property	Type	Description
`spec`	`object`	StrategyDetailsDeployment represents the parsed details of a Deployment InstallStrategy.
`strategy`	`string`	

.spec.install.spec

Description

StrategyDetailsDeployment represents the parsed details of a Deployment InstallStrategy.

Type

`object`

Required

`deployments`

Property	Type	Description
`clusterPermissions`	`array`	
`deployments`	`array`	
`permissions`	`array`	

.spec.install.spec.clusterPermissions

Type

`array`

.spec.install.spec.clusterPermissions[]

Description

StrategyDeploymentPermissions describe the rbac rules and service account needed by the install strategy

Type

`object`

Required

`rules` `serviceAccountName`

Property	Type	Description
`rules`	`array`	
`serviceAccountName`	`string`	

.spec.install.spec.clusterPermissions[].rules

Type

`array`

.spec.install.spec.clusterPermissions[].rules[]

Description

PolicyRule holds information that describes a policy rule, but does not contain information about who the rule applies to or which namespace the rule applies to.

Type

`object`

Required

`verbs`

Property	Type	Description
`apiGroups`	`array`	APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed.
`nonResourceURLs`	`array`	NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.
`resourceNames`	`array`	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
`resources`	`array`	Resources is a list of resources this rule applies to. '*' represents all resources.
`verbs`	`array`	Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule. '*' represents all verbs.

.spec.install.spec.clusterPermissions[].rules[].apiGroups

Description

APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed.

Type

`array`

.spec.install.spec.clusterPermissions[].rules[].apiGroups[]

Type

`string`

.spec.install.spec.clusterPermissions[].rules[].nonResourceURLs

Description

NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.

Type

`array`

.spec.install.spec.clusterPermissions[].rules[].nonResourceURLs[]

Type

`string`

.spec.install.spec.clusterPermissions[].rules[].resourceNames

Description

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

Type

`array`

.spec.install.spec.clusterPermissions[].rules[].resourceNames[]

Type

`string`

.spec.install.spec.clusterPermissions[].rules[].resources

Description

Resources is a list of resources this rule applies to. '*' represents all resources.

Type

`array`

.spec.install.spec.clusterPermissions[].rules[].resources[]

Type

`string`

.spec.install.spec.clusterPermissions[].rules[].verbs

Description

Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule. '*' represents all verbs.

Type

`array`

.spec.install.spec.clusterPermissions[].rules[].verbs[]

Type

`string`

.spec.install.spec.deployments

Type

`array`

.spec.install.spec.deployments[]

Description

StrategyDeploymentSpec contains the name, spec and labels for the deployment ALM should create

Type

`object`

Required

`name` `spec`

Property	Type	Description
`label`	`object`	Set is a map of label:value. It implements Labels.
`name`	`string`	

Property	Type	Description
`spec`	`object`	DeploymentSpec is the specification of the desired behavior of the Deployment.

.spec.install.spec.deployments[].label

Description

Set is a map of label:value. It implements Labels.

Type

`object`

.spec.install.spec.deployments[].spec

Description

DeploymentSpec is the specification of the desired behavior of the Deployment.

Type

`object`

Required

`selector` `template`

Property	Type	Description
`minReadySeconds`	`integer`	Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)

Property	Type	Description
`paused`	`boolean`	Indicates that the deployment is paused.
`progressDeadlineSeconds`	`integer`	The maximum time in seconds for a deployment to make progress before it is considered to be failed. The deployment controller will continue to process failed deployments and a condition with a ProgressDeadlineExceeded reason will be surfaced in the deployment status. Note that progress will not be estimated during the time a deployment is paused. Defaults to 600s.
`replicas`	`integer`	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
`revisionHistoryLimit`	`integer`	The number of old ReplicaSets to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 10.
`selector`	`object`	Label selector for pods. Existing ReplicaSets whose pods are selected by this will be the ones affected by this deployment. It must match the pod template's labels.

Property	Type	Description
`strategy`	`object`	The deployment strategy to use to replace existing pods with new ones.
`template`	`object`	Template describes the pods that will be created.

.spec.install.spec.deployments[].spec.selector

Description

Label selector for pods. Existing ReplicaSets whose pods are selected by this will be the ones affected by this deployment. It must match the pod template's labels.

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.install.spec.deployments[].spec.selector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].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.

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.install.spec.deployments[].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.install.spec.deployments[].spec.selector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].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.install.spec.deployments[].spec.strategy

Description

The deployment strategy to use to replace existing pods with new ones.

Type

`object`

Property	Type	Description
`rollingUpdate`	`object`	Rolling update config params. Present only if DeploymentStrategyType = RollingUpdate. --- TODO: Update this to follow our convention for oneOf, whatever we decide it to be.
`type`	`string`	Type of deployment. Can be "Recreate" or "RollingUpdate". Default is RollingUpdate.

.spec.install.spec.deployments[].spec.strategy.rollingUpdate

Description

Rolling update config params. Present only if DeploymentStrategyType = RollingUpdate. --- TODO: Update this to follow our convention for oneOf, whatever we decide it to be.

Type

`object`

Property	Type	Description
`maxSurge`	`integer|string`	The maximum number of pods that can be scheduled above the desired number of pods. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). This can not be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up. Defaults to 25%. Example: when this is set to 30%, the new ReplicaSet can be scaled up immediately when the rolling update starts, such that the total number of old and new pods do not exceed 130% of desired pods. Once old pods have been killed, new ReplicaSet can be scaled up further, ensuring that total number of pods running at any time during the update is at most 130% of desired pods.
`maxUnavailable`	`integer|string`	The maximum number of pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). Absolute number is calculated from percentage by rounding down. This can not be 0 if MaxSurge is 0. Defaults to 25%. Example: when this is set to 30%, the old ReplicaSet can be scaled down to 70% of desired pods immediately when the rolling update starts. Once new pods are ready, old ReplicaSet can be scaled down further, followed by scaling up the new ReplicaSet, ensuring that the total number of pods available at all times during the update is at least 70% of desired pods.

.spec.install.spec.deployments[].spec.template

Description

Template describes the pods that will be created.

Type

`object`

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

.spec.install.spec.deployments[].spec.template.spec

Description

Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

Type

`object`

Required

`containers`

Property	Type	Description
`activeDeadlineSeconds`	`integer`	Optional duration in seconds the pod may StartTime before the system will actively tr

Property	Type	Description
		containers. Value must be a positive intege
`affinity`	`object`	If specified, the pod's scheduling constrain
`automountServiceAccountToken`	`boolean`	AutomountServiceAccountToken indicates be automatically mounted.
`containers`	`array`	List of containers belonging to the pod. Co removed. There must be at least one conta
`dnsConfig`	`object`	Specifies the DNS parameters of a pod. Pa to the generated DNS configuration based
`dnsPolicy`	`string`	Set DNS policy for the pod. Defaults to "Cl 'ClusterFirstWithHostNet', 'ClusterFirst', 'De in DNSConfig will be merged with the polic DNS options set along with hostNetwork, y to 'ClusterFirstWithHostNet'.
`enableServiceLinks`	`boolean`	EnableServiceLinks indicates whether info injected into pod's environment variables, r Optional: Defaults to true.
`ephemeralContainers`	`array`	List of ephemeral containers run in this poc an existing pod to perform user-initiated ac cannot be specified when creating a pod, a

Property	Type	Description
		the pod spec. In order to add an ephemera pod's ephemeralcontainers subresource. T honored by servers that enable the Ephem
`hostAliases`	`array`	HostAliases is an optional list of hosts and hosts file if specified. This is only valid for r
`hostIPC`	`boolean`	Use the host's ipc namespace. Optional: D
`hostNetwork`	`boolean`	Host networking requested for this pod. Us option is set, the ports that will be used mu
`hostPID`	`boolean`	Use the host's pid namespace. Optional: D
`hostname`	`string`	Specifies the hostname of the Pod If not sp to a system-defined value.
`imagePullSecrets`	`array`	ImagePullSecrets is an optional list of refer namespace to use for pulling any of the im these secrets will be passed to individual p For example, in the case of docker, only D More info: https://kubernetes.io/docs/conce imagepullsecrets-on-a-pod ↗
`initContainers`	`array`	List of initialization containers belonging to order prior to containers being started. If ar

Property	Type	Description
		considered to have failed and is handled a for an init container or normal container mu containers may not have Lifecycle actions, Startup probes. The resourceRequirement account during scheduling by finding the hi type, and then using the max of of that valu Limits are applied to init containers in a sim currently be added or removed. Cannot be https://kubernetes.io/docs/concepts/worklo
`nodeName`	`string`	NodeName is a request to schedule this pc empty, the scheduler simply schedules this fits resource requirements.
`nodeSelector`	`object`	NodeSelector is a selector which must be t which must match a node's labels for the p info: https://kubernetes.io/docs/concepts/c
`overhead`	`object`	Overhead represents the resource overhea given RuntimeClass. This field will be auto RuntimeClass admission controller. If the F enabled, overhead must not be set in Pod admission controller will reject Pod create already set. If RuntimeClass is configured will be set to the value defined in the corre remain unset and treated as zero. More inf https://git.k8s.io/enhancements/keps/sig-n This field is beta-level as of Kubernetes v1 enable the PodOverhead feature.

Property	Type	Description
`preemptionPolicy`	`string`	PreemptionPolicy is the Policy for preempt Never, PreemptLowerPriority. Defaults to P is beta-level, gated by the NonPreemptingI
`priority`	`integer`	The priority value. Various system compon the pod. When Priority Admission Controlle setting this field. The admission controller p PriorityClassName. The higher the value, t
`priorityClassName`	`string`	If specified, indicates the pod's priority. "sy critical" are two special keywords which inc former being the highest priority. Any other PriorityClass object with that name. If not s or zero if there is no default.
`readinessGates`	`array`	If specified, all readiness gates will be eval when all its containers are ready AND all c gates have status equal to "True" More info https://git.k8s.io/enhancements/keps/sig-n
`restartPolicy`	`string`	Restart policy for all containers within the p Default to Always. More info: https://kubernetes.io/docs/concepts/worklo
`runtimeClassName`	`string`	RuntimeClassName refers to a RuntimeCla which should be used to run this pod. If no named class, the pod will not be run. If uns

Property	Type	Description
		will be used, which is an implicit class with default runtime handler. More info: https://g node/585-runtime-class ↗ This is a beta fe
`schedulerName`	`string`	If specified, the pod will be dispatched by s pod will be dispatched by default schedule
`securityContext`	`object`	SecurityContext holds pod-level security at settings. Optional: Defaults to empty. See t each field.
`serviceAccount`	`string`	DeprecatedServiceAccount is a depreciate Deprecated: Use serviceAccountName ins
`serviceAccountName`	`string`	ServiceAccountName is the name of the S More info: https://kubernetes.io/docs/tasks service-account/ ↗
`setHostnameAsFQDN`	`boolean`	If true the pod's hostname will be configure leaf name (the default). In Linux containers hostname field of the kernel (the nodenam containers, this means setting the registry HKEY_LOCAL_MACHINE\SYSTEM\Curre to FQDN. If a pod does not have FQDN, th
`shareProcessNamespace`	`boolean`	Share a single process namespace betwee this is set containers will be able to view ar

Property	Type	Description
		containers in the same pod, and the first pr assigned PID 1. HostPID and ShareProces Optional: Default to false.
`subdomain`	`string`	If specified, the fully qualified Pod hostnam pod will not have a domainname at all.
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need decreased in delete request. Value must b indicates stop immediately via the kill signa value is nil, the default grace period will be duration in seconds after the processes ru signal and the time when the processes ar this value longer than the expected cleanu seconds.
`tolerations`	`array`	If specified, the pod's tolerations.
`topologySpreadConstraints`	`array`	TopologySpreadConstraints describes how across topology domains. Scheduler will sc the constraints. All topologySpreadConstra
`volumes`	`array`	List of volumes that can be mounted by co info: https://kubernetes.io/docs/concepts/st

.spec.install.spec.deployments[].spec.template.spec.affinity

Description

If specified, the pod's scheduling constraints

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.install.spec.deployments[].spec.template.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 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 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

Property	Type	Description
		If the affinity requirem specified by this field cease to be met at sc point during pod execution (e.g. due tc update), the system r or may not try to eventually evict the p from its node.

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`preference` `weight`

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

.spec.install.spec.deployments[].spec.template.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.

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

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

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.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

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

Type

`object`

Required

`key` `operator`

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

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.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE

xecution

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.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

Description

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

Type

`object`

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE

xecution.nodeSelectorTerms[].matchFields[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity

Description

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

Type

`object`

Property	Type	Description
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	The scheduler will 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 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

Property	Type	Description
		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 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 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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`podAffinityTerm` `weight`

Property	Type	Description
`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.install.spec.deployments[].spec.template.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.
`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

Property	Type	Description
		namespace". An empty selector ({}) matches all namespaces. This field is beta-level and is only honored when PodAffinityNamespaceSelector feature is enabled.
`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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

Description

A label query over a set of resources, in this case 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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

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

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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.install.spec.deployments[].spec.template.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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`topologyKey`

Property	Type	Description
`labelSelector`	`object`	A label query over a set of resources, in this case pods.
`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. This field is beta-level and is only honored when PodAffinityNamespaceSelector feature is enabled.
`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

Property	Type	Description
		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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

Description

A label query over a set of resources, in this case 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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

Property	Type	Description
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

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

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.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity

Description

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

Type

`object`

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

Property	Type	Description
		will not be scheduled the node. If the anti-affinity requirements specified by this field cease to be met at sc point during pod execution (e.g. due tc 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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`podAffinityTerm` `weight`

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

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

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

Type

`object`

Required

`topologyKey`

Property	Type	Description
`labelSelector`	`object`	A label query over a set of resources, in this case pods.
`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. This field is beta-level and is only honored when PodAffinityNamespaceSelector feature is enabled.
`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

Property	Type	Description
		that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

Description

A label query over a set of resources, in this case 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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

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

Description

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

Type

`array`

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

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuri

ngExecution[].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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

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

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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.install.spec.deployments[].spec.template.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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuri

ngExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`topologyKey`

Property	Type	Description
`labelSelector`	`object`	A label query over a set of resources, in this case pods.
`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. This field is beta-level and is only honored when PodAffinityNamespaceSelector feature is enabled.
`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

Property	Type	Description
		that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

Description

A label query over a set of resources, in this case 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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

Property	Type	Description
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

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

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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.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.install.spec.deployments[].spec.template.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.install.spec.deployments[].spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[]

Description

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

Type

`object`

Required

`name`

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

Property	Type	Description
`env`	`array`	List of environment variables to set in the contair updated.
`envFrom`	`array`	List of sources to populate environment variables container. The keys defined within a source must C_IDENTIFIER. All invalid keys will be reported a when the container is starting. When a key exists sources, the value associated with the last sourc precedence. Values defined by an Env with a du take precedence. Cannot be updated.
`image`	`string`	Docker image name. More info: https://kubernetes.io/docs/concepts/containers/ir field is optional to allow higher level config mana default or override container images in workload Deployments and StatefulSets.
`imagePullPolicy`	`string`	Image pull policy. One of Always, Never, IfNotPre to Always if :latest tag is specified, or IfNotPrese Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/ir images ↗
`lifecycle`	`object`	Actions that the management system should tak container lifecycle events. Cannot be updated.

Property	Type	Description
`livenessProbe`	`object`	Periodic probe of container liveness. Container w the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗
`name`	`string`	Name of the container specified as a DNS_LABE container in a pod must have a unique name (DN Cannot be updated.
`ports`	`array`	List of ports to expose from the container. Expos gives the system additional information about the connections a container uses, but is primarily inf specifying a port here DOES NOT prevent that p exposed. Any port which is listening on the defau address inside a container will be accessible fror Cannot be updated.
`readinessProbe`	`object`	Periodic probe of container service readiness. C removed from service endpoints if the probe fails updated. More info: https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗
`resources`	`object`	Compute Resources required by this container. C updated. More info: https://kubernetes.io/docs/concepts/configuration resources-containers/ ↗

Property	Type	Description
`securityContext`	`object`	SecurityContext defines the security options the be run with. If set, the fields of SecurityContext o equivalent fields of PodSecurityContext. More inf https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ ↗
`startupProbe`	`object`	StartupProbe indicates that the Pod has success If specified, no other probes are executed until th successfully. If this probe fails, the Pod will be re the livenessProbe failed. This can be used to pro probe parameters at the beginning of a Pod's life might take a long time to load data or warm a ca steady-state operation. This cannot be updated. https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗
`stdin`	`boolean`	Whether this container should allocate a buffer fo container runtime. If this is not set, reads from st container will always result in EOF. Default is fals
`stdinOnce`	`boolean`	Whether the container runtime should close the s after it has been opened by a single attach. Whe the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is oper start, is empty until the first client attaches to stdi remains open and accepts data until the client di which time stdin is closed and remains closed ur is restarted. If this flag is false, a container proce from stdin will never receive an EOF. Default is fa

Property	Type	Description
`terminationMessagePath`	`string`	Optional: Path at which the file to which the conta termination message will be written is mounted i container's filesystem. Message written is intend final status, such as an assertion failure messag truncated by the node if greater than 4096 bytes. message length across all containers will be limit Defaults to /dev/termination-log. Cannot be upda
`terminationMessagePolicy`	`string`	Indicate how the termination message should be will use the contents of terminationMessagePath container status message on both success and f FallbackToLogsOnError will use the last chunk o output if the termination message file is empty ar exited with an error. The log output is limited to 2 lines, whichever is smaller. Defaults to File. Cann
`tty`	`boolean`	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
`volumeDevices`	`array`	volumeDevices is the list of block devices to be u container.
`volumeMounts`	`array`	Pod volumes to mount into the container's filesys updated.
`workingDir`	`string`	Container's working directory. If not specified, the runtime's default will be used, which might be co

Property	Type	Description
		container image. Cannot be updated.

.spec.install.spec.deployments[].spec.template.spec.containers[].args

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].args[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].command

Description

Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be

unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].env

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

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

.spec.install.spec.deployments[].spec.template.spec.containers[].env[].valueFrom

Description

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

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`fieldRef`	`object`	Selects a field of the pod: supports metadata.name, metadata.namespace, `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.install.spec.deployments[].spec.template.spec.containers[].env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.install.spec.deployments[].spec.template.spec.containers[].env[].valueFrom.fieldRef

Description

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

Type

`object`

Required

`fieldPath`

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.containers[].env[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.install.spec.deployments[].spec.template.spec.containers[].env[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key of the secret to select from. Must be a valid secret key.
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.install.spec.deployments[].spec.template.spec.containers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the

last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

`object`

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

.spec.install.spec.deployments[].spec.template.spec.containers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap must be defined

.spec.install.spec.deployments[].spec.template.spec.containers[].envFrom[].secretRef

Description

The Secret to select from

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret must be defined

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle

Description

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

Type

`object`

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

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.postStart

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should be specified. Exec specifies the action to take.
`httpGet`	`object`	HTTPGet specifies the http request to perform.
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.postStart.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.containers[].lifecycle.postStart.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.postStart.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.postStart.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.postStart.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].lifecycle.preStop

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The reason for termination is passed to the handler. The Pod's termination grace period countdown begins before the PreStop hooked is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period. Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should be specified. Exec specifies the action to take.

Property	Type	Description
`httpGet`	`object`	HTTPGet specifies the http request to perform.
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.preStop.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.containers[].lifecycle.preStop.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.preStop.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

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

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`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.containers[].lifecycle.preStop.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

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

Property	Type	Description
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].livenessProbe

Description

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

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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

Property	Type	Description
		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. D Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook
`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 (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.containers[].livenessProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.containers[].livenessProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].livenessProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].livenessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

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

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`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.containers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

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

Property	Type	Description
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].ports

Description

List of ports to expose from the container. Exposing a port here gives the system additional information about the network connections a container uses, but is primarily informational. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Cannot be updated.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

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

.spec.install.spec.deployments[].spec.template.spec.containers[].readinessProbe

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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

Property	Type	Description
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook
`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.install.spec.deployments[].spec.template.spec.containers[].readinessProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.containers[].readinessProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].readinessProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.containers[].readinessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].resources

Description

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

Type

`object`

Property	Type	Description
`limits`	`object`	Limits describes the maximum amount of compute resources allowed. More info:

Property	Type	Description
		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. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.install.spec.deployments[].spec.template.spec.containers[].resources.limits

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.containers[].resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.containers[].securityContext

Description

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

Type

`object`

Property	Type	Description
`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
`capabilities`	`object`	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
`privileged`	`boolean`	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false.

Property	Type	Description
`procMount`	`string`	procMount denotes the type of proc mount to use for the containers. The default is DefaultProcMount which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled.
`readOnlyRootFilesystem`	`boolean`	Whether this container has a read-only root filesystem. Default is false.
`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.
`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.

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 PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
`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.
`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.
`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.

.spec.install.spec.deployments[].spec.template.spec.containers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.

Type

`object`

Property	Type	Description
`add`	`array`	Added capabilities
`drop`	`array`	Removed capabilities

.spec.install.spec.deployments[].spec.template.spec.containers[].securityContext.capabilities.add

Description

Added capabilities

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].securityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].securityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].securityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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.install.spec.deployments[].spec.template.spec.containers[].securityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options.

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

Property	Type	Description
		seccomp profile location. Must only be set if type is "Localhost".
`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.install.spec.deployments[].spec.template.spec.containers[].securityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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.

Property	Type	Description
`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. This field is alpha-level and will only be honored by components that enable the WindowsHostProcessContainers feature flag. Setting this field without the feature flag will result in errors when validating the Pod. 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.install.spec.deployments[].spec.template.spec.containers[].startupProbe

Description

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

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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 TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook
`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 sign value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling 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.install.spec.deployments[].spec.template.spec.containers[].startupProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.containers[].startupProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].startupProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.containers[].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.
`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`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].startupProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.containers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.containers[].volumeDevices

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].volumeDevices[]

Description

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

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
`devicePath`	`string`	devicePath is the path inside of the container that the device will be mapped to.
`name`	`string`	name must match the name of a persistentVolumeClaim in the pod

.spec.install.spec.deployments[].spec.template.spec.containers[].volumeMounts

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.containers[].volumeMounts[]

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.
`name`	`string`	This must match the Name of a Volume.

Property	Type	Description
`readOnly`	`boolean`	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
`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.install.spec.deployments[].spec.template.spec.dnsConfig

Description

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

Type

`object`

Property	Type	Description
`nameservers`	`array`	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from

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

.spec.install.spec.deployments[].spec.template.spec.dnsConfig.nameservers

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.dnsConfig.nameservers[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.dnsConfig.options

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.dnsConfig.options[]

Description

PodDNSConfigOption defines DNS resolver options of a pod.

Type

`object`

Property	Type	Description
`name`	`string`	Required.
`value`	`string`	

.spec.install.spec.deployments[].spec.template.spec.dnsConfig.searches

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.dnsConfig.searches[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[]

Description

An EphemeralContainer is a container that may be added temporarily to an existing pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a pod is removed or restarted. If an ephemeral container causes a pod to exceed its resource allocation, the pod may be evicted. Ephemeral containers may not be added by directly updating the pod spec. They must be added via the pod's ephemeralcontainers subresource, and they will appear in the pod spec once added. This is an alpha feature enabled by the EphemeralContainers feature flag.

Type

`object`

Required

`name`

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

Property	Type	Description
`env`	`array`	List of environment variables to set in the contair updated.
`envFrom`	`array`	List of sources to populate environment variables container. The keys defined within a source must C_IDENTIFIER. All invalid keys will be reported a when the container is starting. When a key exists sources, the value associated with the last sourc precedence. Values defined by an Env with a du take precedence. Cannot be updated.
`image`	`string`	Docker image name. More info: https://kubernetes.io/docs/concepts/containers/ir
`imagePullPolicy`	`string`	Image pull policy. One of Always, Never, IfNotPre to Always if :latest tag is specified, or IfNotPresei Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/ir images ↗
`lifecycle`	`object`	Lifecycle is not allowed for ephemeral containers
`livenessProbe`	`object`	Probes are not allowed for ephemeral containers
`name`	`string`	Name of the ephemeral container specified as a This name must be unique among all containers,

Property	Type	Description
		and ephemeral containers.
`ports`	`array`	Ports are not allowed for ephemeral containers.
`readinessProbe`	`object`	Probes are not allowed for ephemeral containers
`resources`	`object`	Resources are not allowed for ephemeral contai containers use spare resources already allocate
`securityContext`	`object`	Optional: SecurityContext defines the security op ephemeral container should be run with. If set, th SecurityContext override the equivalent fields of PodSecurityContext.
`startupProbe`	`object`	Probes are not allowed for ephemeral containers
`stdin`	`boolean`	Whether this container should allocate a buffer fo container runtime. If this is not set, reads from st container will always result in EOF. Default is fals
`stdinOnce`	`boolean`	Whether the container runtime should close the s after it has been opened by a single attach. Whe the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is oper start, is empty until the first client attaches to stdi remains open and accepts data until the client di

Property	Type	Description
		which time stdin is closed and remains closed ur is restarted. If this flag is false, a container proce from stdin will never receive an EOF. Default is fa
`targetContainerName`	`string`	If set, the name of the container from PodSpec th ephemeral container targets. The ephemeral cor run in the namespaces (IPC, PID, etc) of this cor then the ephemeral container is run in whatever shared for the pod. Note that the container runtin this feature.
`terminationMessagePath`	`string`	Optional: Path at which the file to which the conta termination message will be written is mounted in container's filesystem. Message written is intend final status, such as an assertion failure messag truncated by the node if greater than 4096 bytes. message length across all containers will be limit Defaults to /dev/termination-log. Cannot be upda
`terminationMessagePolicy`	`string`	Indicate how the termination message should be will use the contents of terminationMessagePath container status message on both success and f FallbackToLogsOnError will use the last chunk of output if the termination message file is empty an exited with an error. The log output is limited to 2 lines, whichever is smaller. Defaults to File. Cann
`tty`	`boolean`	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.

Property	Type	Description
`volumeDevices`	`array`	volumeDevices is the list of block devices to be u container.
`volumeMounts`	`array`	Pod volumes to mount into the container's filesys updated.
`workingDir`	`string`	Container's working directory. If not specified, the runtime's default will be used, which might be co container image. Cannot be updated.

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].args

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].args[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].env

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

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

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].env[].valueFrom

Description

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

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].env[].valueFrom.fieldRef

Description

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

Type

`object`

Required

`fieldPath`

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

Property	Type	Description
`containerName`	`string`	Container name: required for volumes, optional for env vars

Property	Type	Description
`divisor`	`integer|string`	Specifies the output format of the exposed resources, defaults to "1"
`resource`	`string`	Required: resource to select

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key of the secret to select from. Must be a valid secret key.
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?

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

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].envFrom

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

`object`

Property	Type	Description
`configMapRef`	`object`	The ConfigMap to select from

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

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap must be defined

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].envFrom[].secretRef

Description

The Secret to select from

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret must be defined

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle

Description

Lifecycle is not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
`postStart`	`object`	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗

Property	Type	Description
`preStop`	`object`	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The reason for termination is passed to the handler. The Pod's termination grace period countdown begins before the PreStop hooked is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period. Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.postStart

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should be specified. Exec specifies the action to take.
`httpGet`	`object`	HTTPGet specifies the http request to perform.
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[

]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.preStop

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The reason for termination is passed to the handler. The Pod's termination grace period countdown begins before the PreStop hooked is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period. Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should be specified. Exec specifies the action to take.

Property	Type	Description
`httpGet`	`object`	HTTPGet specifies the http request to perform.
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

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

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`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.preStop.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
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

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

Property	Type	Description
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].livenessProbe

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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. D Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook
`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 (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP

Property	Type	Description
		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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].livenessProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].livenessProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].livenessProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].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.

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`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

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

Property	Type	Description
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].ports

Description

Ports are not allowed for ephemeral containers.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

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

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

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].readinessProbe

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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. D Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook

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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].readinessProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].readinessProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].readinessProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].readinessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].resources

Description

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

Type

`object`

Property	Type	Description
`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

Property	Type	Description
		an implementation-defined value. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].resources.limits

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext

Description

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

PodSecurityContext.

Type

`object`

Property	Type	Description
`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
`capabilities`	`object`	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
`privileged`	`boolean`	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false.
`procMount`	`string`	procMount denotes the type of proc mount to use for the containers. The default is DefaultProcMount which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled.

Property	Type	Description
`readOnlyRootFilesystem`	`boolean`	Whether this container has a read-only root filesystem. Default is false.
`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.
`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.

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.
`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.
`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.

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.

Type

`object`

Property	Type	Description
`add`	`array`	Added capabilities
`drop`	`array`	Removed capabilities

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext.capabilities.add

Description

Added capabilities

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options.

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 only be set if type is "Localhost".

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].securityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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. This field is alpha-level and will only be honored by components that enable the WindowsHostProcessContainers feature flag. Setting this field without the feature flag will result in errors when validating the Pod. 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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].startupProbe

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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. D Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook

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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].startupProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].startupProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].startupProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].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.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.ephemeralContainers[].volumeDevices

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].volumeDevices[]

Description

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

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
`devicePath`	`string`	devicePath is the path inside of the container that the device will be mapped to.
`name`	`string`	name must match the name of a persistentVolumeClaim in the pod

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].volumeMounts

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.ephemeralContainers[].volumeMounts[]

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.
`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.
`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.install.spec.deployments[].spec.template.spec.hostAliases

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified. This is only valid for non-hostNetwork pods.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.hostAliases[]

Description

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

Type

`object`

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

.spec.install.spec.deployments[].spec.template.spec.hostAliases[].hostnames

Description

Hostnames for the above IP address.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.hostAliases[].hostnames[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.imagePullSecrets

Description

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. For example, in the case of docker, only DockerConfig type secrets are honored. More info: https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.imagePullSecrets[]

Description

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

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.initContainers

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[]

Description

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

Type

`object`

Required

`name`

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

Property	Type	Description
`envFrom`	`array`	List of sources to populate environment variables container. The keys defined within a source must C_IDENTIFIER. All invalid keys will be reported a when the container is starting. When a key exists sources, the value associated with the last sourc precedence. Values defined by an Env with a du take precedence. Cannot be updated.
`image`	`string`	Docker image name. More info: https://kubernetes.io/docs/concepts/containers/ir field is optional to allow higher level config mana default or override container images in workload Deployments and StatefulSets.
`imagePullPolicy`	`string`	Image pull policy. One of Always, Never, IfNotPre to Always if :latest tag is specified, or IfNotPrese Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/ir images ↗
`lifecycle`	`object`	Actions that the management system should tak container lifecycle events. Cannot be updated.
`livenessProbe`	`object`	Periodic probe of container liveness. Container w the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗

Property	Type	Description
`name`	`string`	Name of the container specified as a DNS_LABE container in a pod must have a unique name (DN Cannot be updated.
`ports`	`array`	List of ports to expose from the container. Expos gives the system additional information about the connections a container uses, but is primarily inf specifying a port here DOES NOT prevent that p exposed. Any port which is listening on the defau address inside a container will be accessible fror Cannot be updated.
`readinessProbe`	`object`	Periodic probe of container service readiness. C removed from service endpoints if the probe fails updated. More info: https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗
`resources`	`object`	Compute Resources required by this container. C updated. More info: https://kubernetes.io/docs/concepts/configuratior resources-containers/ ↗

Property	Type	Description
`securityContext`	`object`	SecurityContext defines the security options the be run with. If set, the fields of SecurityContext o equivalent fields of PodSecurityContext. More inf https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ ↗
`startupProbe`	`object`	StartupProbe indicates that the Pod has success If specified, no other probes are executed until th successfully. If this probe fails, the Pod will be re the livenessProbe failed. This can be used to pro probe parameters at the beginning of a Pod's life might take a long time to load data or warm a ca steady-state operation. This cannot be updated. https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗
`stdin`	`boolean`	Whether this container should allocate a buffer fo container runtime. If this is not set, reads from st container will always result in EOF. Default is fals
`stdinOnce`	`boolean`	Whether the container runtime should close the s after it has been opened by a single attach. Whe the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is oper start, is empty until the first client attaches to stdi remains open and accepts data until the client di which time stdin is closed and remains closed ur is restarted. If this flag is false, a container proce from stdin will never receive an EOF. Default is fa

Property	Type	Description
`terminationMessagePath`	`string`	Optional: Path at which the file to which the conta termination message will be written is mounted i container's filesystem. Message written is intend final status, such as an assertion failure messag truncated by the node if greater than 4096 bytes. message length across all containers will be limit Defaults to /dev/termination-log. Cannot be upda
`terminationMessagePolicy`	`string`	Indicate how the termination message should be will use the contents of terminationMessagePath container status message on both success and f FallbackToLogsOnError will use the last chunk o output if the termination message file is empty ar exited with an error. The log output is limited to 2 lines, whichever is smaller. Defaults to File. Cann
`tty`	`boolean`	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
`volumeDevices`	`array`	volumeDevices is the list of block devices to be u container.
`volumeMounts`	`array`	Pod volumes to mount into the container's filesys updated.
`workingDir`	`string`	Container's working directory. If not specified, the runtime's default will be used, which might be co

Property	Type	Description
		container image. Cannot be updated.

.spec.install.spec.deployments[].spec.template.spec.initContainers[].args

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].args[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].command

Description

Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be

unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].env

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

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

.spec.install.spec.deployments[].spec.template.spec.initContainers[].env[].valueFrom

Description

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

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`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.install.spec.deployments[].spec.template.spec.initContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.install.spec.deployments[].spec.template.spec.initContainers[].env[].valueFrom.fieldRef

Description

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

Type

`object`

Required

`fieldPath`

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.initContainers[].env[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.install.spec.deployments[].spec.template.spec.initContainers[].env[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key of the secret to select from. Must be a valid secret key.
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.install.spec.deployments[].spec.template.spec.initContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the

last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

`object`

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

.spec.install.spec.deployments[].spec.template.spec.initContainers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap must be defined

.spec.install.spec.deployments[].spec.template.spec.initContainers[].envFrom[].secretRef

Description

The Secret to select from

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret must be defined

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle

Description

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

Type

`object`

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

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.postStart

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should be specified. Exec specifies the action to take.
`httpGet`	`object`	HTTPGet specifies the http request to perform.
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.postStart.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.postStart.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.postStart.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.postStart.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.preStop

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The reason for termination is passed to the handler. The Pod's termination grace period countdown begins before the PreStop hooked is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period. Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should be specified. Exec specifies the action to take.

Property	Type	Description
`httpGet`	`object`	HTTPGet specifies the http request to perform.
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.preStop.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.preStop.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.preStop.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

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

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`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.initContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

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

Property	Type	Description
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].livenessProbe

Description

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

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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

Property	Type	Description
		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. D Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook
`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 (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.initContainers[].livenessProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.initContainers[].livenessProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].livenessProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].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.

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`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

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

Property	Type	Description
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].ports

Description

List of ports to expose from the container. Exposing a port here gives the system additional information about the network connections a container uses, but is primarily informational. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Cannot be updated.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

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

.spec.install.spec.deployments[].spec.template.spec.initContainers[].readinessProbe

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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

Property	Type	Description
`tcpSocket`	`object`	TCPSocket specifies an action involving a TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook
`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.install.spec.deployments[].spec.template.spec.initContainers[].readinessProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.initContainers[].readinessProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].readinessProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].readinessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.initContainers[].readinessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].resources

Description

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

Type

`object`

Property	Type	Description
`limits`	`object`	Limits describes the maximum amount of compute resources allowed. More info:

Property	Type	Description
		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. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.install.spec.deployments[].spec.template.spec.initContainers[].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.install.spec.deployments[].spec.template.spec.initContainers[].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. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].securityContext

Description

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

Type

`object`

Property	Type	Description
`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
`capabilities`	`object`	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
`privileged`	`boolean`	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false.

Property	Type	Description
`procMount`	`string`	procMount denotes the type of proc mount to use for the containers. The default is DefaultProcMount which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled.
`readOnlyRootFilesystem`	`boolean`	Whether this container has a read-only root filesystem. Default is false.
`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.
`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.

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 PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
`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.
`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.
`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.

.spec.install.spec.deployments[].spec.template.spec.initContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.

Type

`object`

Property	Type	Description
`add`	`array`	Added capabilities
`drop`	`array`	Removed capabilities

.spec.install.spec.deployments[].spec.template.spec.initContainers[].securityContext.capabilities.add

Description

Added capabilities

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].securityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].securityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].securityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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.install.spec.deployments[].spec.template.spec.initContainers[].securityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options.

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

Property	Type	Description
		seccomp profile location. Must only be set if type is "Localhost".
`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.install.spec.deployments[].spec.template.spec.initContainers[].securityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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.

Property	Type	Description
`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. This field is alpha-level and will only be honored by components that enable the WindowsHostProcessContainers feature flag. Setting this field without the feature flag will result in errors when validating the Pod. 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.install.spec.deployments[].spec.template.spec.initContainers[].startupProbe

Description

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

Type

`object`

Property	Type	Description
`exec`	`object`	One and only one of the following should b Exec specifies the action to take.
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`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 TCP hooks not yet supported TODO: imple realistic TCP lifecycle hook
`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 sign value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling 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.install.spec.deployments[].spec.template.spec.initContainers[].startupProbe.exec

Description

One and only one of the following should be specified. 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.install.spec.deployments[].spec.template.spec.initContainers[].startupProbe.exec.command

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].startupProbe.exec.command[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].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.
`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`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name
`value`	`string`	The header field value

.spec.install.spec.deployments[].spec.template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported TODO: implement a realistic TCP lifecycle hook

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	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.install.spec.deployments[].spec.template.spec.initContainers[].volumeDevices

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].volumeDevices[]

Description

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

Type

`object`

Required

`devicePath` `name`

Property	Type	Description
`devicePath`	`string`	devicePath is the path inside of the container that the device will be mapped to.
`name`	`string`	name must match the name of a persistentVolumeClaim in the pod

.spec.install.spec.deployments[].spec.template.spec.initContainers[].volumeMounts

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.initContainers[].volumeMounts[]

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.
`name`	`string`	This must match the Name of a Volume.

Property	Type	Description
`readOnly`	`boolean`	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
`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.install.spec.deployments[].spec.template.spec.nodeSelector

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.overhead

Description

Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md This field is beta-level as of Kubernetes v1.18, and is only honored by servers that enable the PodOverhead feature.

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.readinessGates

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.readinessGates[]

Description

PodReadinessGate contains the reference to a pod condition

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	ConditionType refers to a condition in the pod's condition list with matching type.

.spec.install.spec.deployments[].spec.template.spec.securityContext

Description

SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See type description for default values of each field.

Type

`object`

Property	Type	Description
`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.
`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.
`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.
`runAsNonRoot`	`boolean`	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
`runAsUser`	`integer`	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in

Property	Type	Description
		SecurityContext takes precedence for that container.
`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.
`seccompProfile`	`object`	The seccomp options to use by the containers in this pod.
`supplementalGroups`	`array`	A list of groups applied to the first process run in each container, in addition to the container's primary GID. If unspecified, no groups will be added to any container.
`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.
`windowsOptions`	`object`	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext,

Property	Type	Description
		the value specified in SecurityContext takes precedence.

.spec.install.spec.deployments[].spec.template.spec.securityContext.seLinuxOptions

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.

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.install.spec.deployments[].spec.template.spec.securityContext.seccompProfile

Description

The seccomp options to use by the containers in this pod.

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 only be set if type is "Localhost".
`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.install.spec.deployments[].spec.template.spec.securityContext.supplementalGroups

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID. If unspecified, no groups will be added to any container.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.securityContext.supplementalGroups[]

Type

`integer`

.spec.install.spec.deployments[].spec.template.spec.securityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	Name of a property to set

Property	Type	Description
`value`	`string`	Value of a property to set

.spec.install.spec.deployments[].spec.template.spec.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.

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. This field is alpha-level and will only be honored by components that enable the

Property	Type	Description
		WindowsHostProcessContainers feature flag. Setting this field without the feature flag will result in errors when validating the Pod. 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.install.spec.deployments[].spec.template.spec.tolerations

Description

If specified, the pod's tolerations.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
`effect`	`string`	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, 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.

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.topologySpreadConstraints

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.topologySpreadConstraints[]

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

`object`

Required

`maxSkew` `topologyKey` `whenUnsatisfiable`

Property	Type	Description
`labelSelector`	`object`	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
`maxSkew`	`integer`	MaxSkew describes the degree to which pods may be unevenly distributed. When `whenUnsatisfiable=DoNotSchedule`, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 1/1/0: | zone1 | zone2 | zone3 | | P | P | | - if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 1/1/1; scheduling it onto zone1(zone2) would make the ActualSkew(2-0) on zone1(zone2) violate MaxSkew(1). - if MaxSkew is 2, incoming pod can be scheduled onto any zone. When `whenUnsatisfiable=ScheduleAnyway`, it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.
`topologyKey`	`string`	TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket. It's a required field.

Property	Type	Description
`whenUnsatisfiable`	`string`	WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. - DoNotSchedule (default) tells the scheduler not to schedule it. - ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assigment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: | zone1 | zone2 | zone3 | | P P P | P | P | If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it *more* imbalanced. It's a required field.

.spec.install.spec.deployments[].spec.template.spec.topologySpreadConstraints[].labelSelector

Description

LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.

Type

`object`

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.install.spec.deployments[].spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.topologySpreadConstraints[].labelSelector.matchLabels

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.volumes

Description

List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[]

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

`object`

Required

`name`

Property	Type	Description
`awsElasticBlockStore`	`object`	AWSElasticBlockStore represents an AWS Disk resc to a kubelet's host machine and then exposed to the https://kubernetes.io/docs/concepts/storage/volumes ↗
`azureDisk`	`object`	AzureDisk represents an Azure Data Disk mount on mount to the pod.
`azureFile`	`object`	AzureFile represents an Azure File Service mount or mount to the pod.
`cephfs`	`object`	CephFS represents a Ceph FS mount on the host th lifetime
`cinder`	`object`	Cinder represents a cinder volume attached and mo host machine. More info: https://examples.k8s.io/my pd/README.md ↗
`configMap`	`object`	ConfigMap represents a configMap that should popu
`csi`	`object`	CSI (Container Storage Interface) represents ephem handled by certain external CSI drivers (Beta feature

Property	Type	Description
`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 the volume is only needed while the pod runs, b) fea volumes like restoring from snapshot or capacity trac the storage driver is specified through a storage clas driver supports dynamic volume provisioning throug PersistentVolumeClaim (see EphemeralVolumeSour information on the connection between this volume t PersistentVolumeClaim). Use PersistentVolumeClair vendor-specific APIs for volumes that persist for long an individual pod. Use CSI for light-weight local ephe CSI driver is meant to be used that way - see the do driver for more information. A pod can use both type volumes and persistent volumes at the same time. T and only available when the GenericEphemeralVolu enabled.
`fc`	`object`	FC represents a Fibre Channel resource that is attac host machine and then exposed to the pod.

Property	Type	Description
`flexVolume`	`object`	FlexVolume represents a generic volume resource tl provisioned/attached using an exec based plugin.
`flocker`	`object`	Flocker represents a Flocker volume attached to a k 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
`gitRepo`	`object`	GitRepo represents a git repository at a particular re DEPRECATED: GitRepo is deprecated. To provision repo, mount an EmptyDir into an InitContainer that c git, then mount the EmptyDir into the Pod's containe
`glusterfs`	`object`	Glusterfs represents a Glusterfs mount on the host t lifetime. More info: https://examples.k8s.io/volumes/ ↗
`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 TODO(jonesdl) We need to restrict who can use hos and who can/can not mount host directories as read

Property	Type	Description
`iscsi`	`object`	ISCSI represents an ISCSI Disk resource that is atta host machine and then exposed to the pod. More inf https://examples.k8s.io/volumes/iscsi/README.md ↗
`name`	`string`	Volume's name. Must be a DNS_LABEL and unique info: https://kubernetes.io/docs/concepts/overview/w objects/names/#names ↗
`nfs`	`object`	NFS represents an NFS mount on the host that shar More info: https://kubernetes.io/docs/concepts/stora
`persistentVolumeClaim`	`object`	PersistentVolumeClaimVolumeSource represents a PersistentVolumeClaim in the same namespace. Mo https://kubernetes.io/docs/concepts/storage/persiste volumes#persistentvolumeclaims ↗
`photonPersistentDisk`	`object`	PhotonPersistentDisk represents a PhotonControlle attached and mounted on kubelets host machine
`portworxVolume`	`object`	PortworxVolume represents a portworx volume attac kubelets host machine
`projected`	`object`	Items for all in one resources secrets, configmaps, a

Property	Type	Description
`quobyte`	`object`	Quobyte represents a Quobyte mount on the host th 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
`storageos`	`object`	StorageOS represents a StorageOS volume attache Kubernetes nodes.
`vsphereVolume`	`object`	VsphereVolume represents a vSphere volume attacl kubelets host machine

.spec.install.spec.deployments[].spec.template.spec.volumes[].awsElasticBlockStore

Description

AWSElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info:

https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	Filesystem type of the volume that you want to mount. Tip: Ensu filesystem type is supported by the host operating system. Exan "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗ TODO: how do we prevent errors in the filesystem from comp the machine
`partition`	`integer`	The partition in the volume that you want to mount. If omitted, th is to mount by volume name. Examples: For volume /dev/sda1, specify the partition as "1". Similarly, the volume partition for /de "0" (or you can leave the property empty).
`readOnly`	`boolean`	Specify "true" to force and set the ReadOnly property in Volume "true". If omitted, the default is "false". More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗
`volumeID`	`string`	Unique ID of the persistent disk resource in AWS (Amazon EBS More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗

.spec.install.spec.deployments[].spec.template.spec.volumes[].azureDisk

Description

AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

`object`

Required

`diskName` `diskURI`

Property	Type	Description
`cachingMode`	`string`	Host Caching mode: None, Read Only, Read Write.
`diskName`	`string`	The Name of the data disk in the blob storage
`diskURI`	`string`	The URI the data disk in the blob storage
`fsType`	`string`	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`	Expected values Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

Property	Type	Description
`readOnly`	`boolean`	Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

.spec.install.spec.deployments[].spec.template.spec.volumes[].azureFile

Description

AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

Type

`object`

Required

`secretName` `shareName`

Property	Type	Description
`readOnly`	`boolean`	Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretName`	`string`	the name of secret that contains Azure Storage Account Name and Key
`shareName`	`string`	Share Name

.spec.install.spec.deployments[].spec.template.spec.volumes[].cephfs

Description

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

Type

`object`

Required

`monitors`

Property	Type	Description
`monitors`	`array`	Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`path`	`string`	Optional: Used as the mounted root, rather than the full Ceph tree, default is /
`readOnly`	`boolean`	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`	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`	Optional: SecretRef is reference to the authentication secret for User, default is empty. More info:

Property	Type	Description
		https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`user`	`string`	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.install.spec.deployments[].spec.template.spec.volumes[].cephfs.monitors

Description

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.install.spec.deployments[].spec.template.spec.volumes[].cephfs.monitors[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.volumes[].cephfs.secretRef

Description

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. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.volumes[].cinder

Description

Cinder represents a cinder volume attached and mounted on kubelets host machine. More info: https://examples.k8s.io/mysql-cinder-pd/README.md

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	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 ↗

Property	Type	Description
`readOnly`	`boolean`	Optional: 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`	Optional: points to a secret object containing parameters used to connect to OpenStack.
`volumeID`	`string`	volume id used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

.spec.install.spec.deployments[].spec.template.spec.volumes[].cinder.secretRef

Description

Optional: points to a secret object containing parameters used to connect to OpenStack.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.volumes[].configMap

Description

ConfigMap represents a configMap that should populate this volume

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	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`	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. More info: https://kubernetes.io/docs/concepts/overview/working-with-

Property	Type	Description
		objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap or its keys must be defined

.spec.install.spec.deployments[].spec.template.spec.volumes[].configMap.items

Description

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.install.spec.deployments[].spec.template.spec.volumes[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	The key to project.
`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`	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.install.spec.deployments[].spec.template.spec.volumes[].csi

Description

CSI (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Type

`object`

Required

`driver`

Property	Type	Description
`driver`	`string`	Driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
`fsType`	`string`	Filesystem type 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`	Specifies a read-only configuration for the volume. Defaults to false (read/write).
`volumeAttributes`	`object`	VolumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

.spec.install.spec.deployments[].spec.template.spec.volumes[].csi.nodePublishSecretRef

Description

NodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.volumes[].csi.volumeAttributes

Description

VolumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.volumes[].downwardAPI

Description

DownwardAPI represents downward API about the pod that should populate this volume

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`items`	`array`	Items is a list of downward API volume file

.spec.install.spec.deployments[].spec.template.spec.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[].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 and namespace are supported.
`mode`	`integer`	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`path`	`string`	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.volumes[].downwardAPI.items[].fieldRef

Description

Required: Selects a field of the pod: only annotations, labels, name and namespace 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.install.spec.deployments[].spec.template.spec.volumes[].downwardAPI.items[].resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.install.spec.deployments[].spec.template.spec.volumes[].emptyDir

Description

EmptyDir represents a temporary directory that shares a pod's lifetime. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Type

`object`

Property	Type	Description
`medium`	`string`	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#empty ↗
`sizeLimit`	`integer|string`	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 b 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: http://kubernetes.io/docs/user-guide/volumes#emptydir

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral

Description

Ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual pod. Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time. This is a beta feature and only available when the GenericEphemeralVolume feature gate is enabled.

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.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate

Description

Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be `<pod name>-<volume name>` where `<volume name>` is the name from the `PodSpec.Volumes` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

Type

`object`

Required

`spec`

Property	Type	Description
`metadata`	`ObjectMeta`	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.
`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.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

Description

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

Type

`object`

Property	Type	Description
`accessModes`	`array`	AccessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1 ↗
`dataSource`	`object`	This 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. If the AnyVolumeDataSource feature gate is enabled, this field will always have the same contents as the DataSourceRef field.
`dataSourceRef`	`object`	Specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any local 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-

Property	Type	Description
		empty, they must have the same value. For backwards compatibility, 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. There are two 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. (Alpha) Using this field requires the AnyVolumeDataSource feature gate to be enabled.
`resources`	`object`	Resources represents the minimum resources the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources ↗
`selector`	`object`	A label query over volumes to consider for binding.
`storageClassName`	`string`	Name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1 ↗

Property	Type	Description
`volumeMode`	`string`	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.
`volumeName`	`string`	VolumeName is the binding reference to the PersistentVolume backing this claim.

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes

Description

AccessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

Description

This 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. If the AnyVolumeDataSource feature gate is enabled, this field will always have the same contents as the DataSourceRef field.

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.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef

Description

Specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any local 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, 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. There are two 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. (Alpha) Using this field requires the AnyVolumeDataSource feature gate to be enabled.

Type

`object`

Required

`kind` `name`

Property	Type	Description
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

Resources represents the minimum resources the volume should have. 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. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector

Description

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.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

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.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values

Description

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

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels

Description

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

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.volumes[].fc

Description

FC represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Type

`object`

Property	Type	Description
`fsType`	`string`	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. TODO: how do we prevent errors in the filesystem from compromising the machine
`lun`	`integer`	Optional: FC target lun number
`readOnly`	`boolean`	Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`targetWWNs`	`array`	Optional: FC target worldwide names (WWNs)

Property	Type	Description
`wwids`	`array`	Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

.spec.install.spec.deployments[].spec.template.spec.volumes[].fc.targetWWNs

Description

Optional: FC target worldwide names (WWNs)

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[].fc.targetWWNs[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.volumes[].fc.wwids

Description

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.install.spec.deployments[].spec.template.spec.volumes[].fc.wwids[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.volumes[].flexVolume

Description

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

Type

`object`

Required

`driver`

Property	Type	Description
`driver`	`string`	Driver is the name of the driver to use for this volume.
`fsType`	`string`	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`	Optional: Extra command options if any.

Property	Type	Description
`readOnly`	`boolean`	Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretRef`	`object`	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.install.spec.deployments[].spec.template.spec.volumes[].flexVolume.options

Description

Optional: Extra command options if any.

Type

`object`

.spec.install.spec.deployments[].spec.template.spec.volumes[].flexVolume.secretRef

Description

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. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.volumes[].flocker

Description

Flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Type

`object`

Property	Type	Description
`datasetName`	`string`	Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
`datasetUUID`	`string`	UUID of the dataset. This is unique identifier of a Flocker dataset

.spec.install.spec.deployments[].spec.template.spec.volumes[].gcePersistentDisk

Description

GCEPersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Type

`object`

Required

`pdName`

Property	Type	Description
`fsType`	`string`	Filesystem type of the volume that you want to mount. Tip: Ensu 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#gcepersis ↗ TODO: how do we prevent errors in the filesystem from compromising the machine
`partition`	`integer`	The partition in the volume that you want to mount. If omitted, th default is to mount by volume name. Examples: For volume /dev you specify the partition as "1". Similarly, the volume partition fo /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗
`pdName`	`string`	Unique name of the PD resource in GCE. Used to identify the di GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗

Property	Type	Description
`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.install.spec.deployments[].spec.template.spec.volumes[].gitRepo

Description

GitRepo represents a git repository at a particular revision. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
`directory`	`string`	Target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
`repository`	`string`	Repository URL

Property	Type	Description
`revision`	`string`	Commit hash for the specified revision.

.spec.install.spec.deployments[].spec.template.spec.volumes[].glusterfs

Description

Glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/glusterfs/README.md

Type

`object`

Required

`endpoints` `path`

Property	Type	Description
`endpoints`	`string`	EndpointsName 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-a-pod ↗
`readOnly`	`boolean`	ReadOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info:

Property	Type	Description
		https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗

.spec.install.spec.deployments[].spec.template.spec.volumes[].hostPath

Description

HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath --- TODO(jonesdl) We need to restrict who can use host directory mounts and who can/can not mount host directories as read/write.

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

.spec.install.spec.deployments[].spec.template.spec.volumes[].iscsi

Description

ISCSI represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://examples.k8s.io/volumes/iscsi/README.md

Type

`object`

Required

`iqn` `lun` `targetPortal`

Property	Type	Description
`chapAuthDiscovery`	`boolean`	whether support iSCSI Discovery CHAP authentication
`chapAuthSession`	`boolean`	whether support iSCSI Session CHAP authentication
`fsType`	`string`	Filesystem type of the volume that you want to mount. Ti Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitl inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#isc ↗ TODO: how do we prevent errors in the filesystem fror compromising the machine
`initiatorName`	`string`	Custom iSCSI Initiator Name. If initiatorName is specifiec with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.

Property	Type	Description
`iqn`	`string`	Target iSCSI Qualified Name.
`iscsiInterface`	`string`	iSCSI Interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
`lun`	`integer`	iSCSI Target Lun number.
`portals`	`array`	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`	CHAP Secret for iSCSI target and initiator authentication
`targetPortal`	`string`	iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

.spec.install.spec.deployments[].spec.template.spec.volumes[].iscsi.portals

Description

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.install.spec.deployments[].spec.template.spec.volumes[].iscsi.portals[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.volumes[].iscsi.secretRef

Description

CHAP Secret for iSCSI target and initiator authentication

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.volumes[].nfs

Description

NFS represents an NFS mount on the host that shares a pod's lifetime More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

Type

`object`

Required

`path` `server`

Property	Type	Description
`path`	`string`	Path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗
`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.install.spec.deployments[].spec.template.spec.volumes[].persistentVolumeClaim

Description

PersistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Type

`object`

Required

`claimName`

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

.spec.install.spec.deployments[].spec.template.spec.volumes[].photonPersistentDisk

Description

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

Type

`object`

Required

`pdID`

Property	Type	Description
`fsType`	`string`	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
`pdID`	`string`	ID that identifies Photon Controller persistent disk

.spec.install.spec.deployments[].spec.template.spec.volumes[].portworxVolume

Description

PortworxVolume represents a portworx volume attached and mounted on kubelets host machine

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	FSType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
`readOnly`	`boolean`	Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`volumeID`	`string`	VolumeID uniquely identifies a Portworx volume

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected

Description

Items for all in one resources secrets, configmaps, and downward API

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	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`	list of volume projections

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources

Description

list of volume projections

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[]

Description

Projection that may be projected along with other supported volume types

Type

`object`

Property	Type	Description
`configMap`	`object`	information about the configMap data to project
`downwardAPI`	`object`	information about the downwardAPI data to project
`secret`	`object`	information about the secret data to project
`serviceAccountToken`	`object`	information about the serviceAccountToken data to project

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].configMap

Description

information about the configMap data to project

Type

`object`

Property	Type	Description
`items`	`array`	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. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap or its keys must be defined

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].configMap.items

Description

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.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	The key to project.
`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`	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.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].downwardAPI

Description

information about the downwardAPI data to project

Type

`object`

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

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].downwardAPI.items

Description

Items is a list of DownwardAPIVolume file

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

`object`

Required

`path`

Property	Type	Description
`fieldRef`	`object`	Required: Selects a field of the pod: only annotations, labels, name and namespace are supported.
`mode`	`integer`	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`path`	`string`	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef

Description

Required: Selects a field of the pod: only annotations, labels, name and namespace 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.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].secret

Description

information about the secret data to project

Type

`object`

Property	Type	Description
`items`	`array`	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 '..'.

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].secret.items

Description

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.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	The key to project.
`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`	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.install.spec.deployments[].spec.template.spec.volumes[].projected.sources[].serviceAccountToken

Description

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.install.spec.deployments[].spec.template.spec.volumes[].quobyte

Description

Quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Type

`object`

Required

`registry` `volume`

Property	Type	Description
`group`	`string`	Group to map volume access to Default is no group
`readOnly`	`boolean`	ReadOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
`registry`	`string`	Registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
`tenant`	`string`	Tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
`user`	`string`	User to map volume access to Defaults to serivceaccount user
`volume`	`string`	Volume is a string that references an already created Quobyte volume by name.

.spec.install.spec.deployments[].spec.template.spec.volumes[].rbd

Description

RBD represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/rbd/README.md

Type

`object`

Required

`image` `monitors`

Property	Type	Description
`fsType`	`string`	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 ↗ TODO: how do we prevent errors in the filesystem from compromising the machine
`image`	`string`	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`	A collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

Property	Type	Description
`pool`	`string`	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`	The rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

.spec.install.spec.deployments[].spec.template.spec.volumes[].rbd.monitors

Description

A collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Type

`array`

.spec.install.spec.deployments[].spec.template.spec.volumes[].rbd.monitors[]

Type

`string`

.spec.install.spec.deployments[].spec.template.spec.volumes[].rbd.secretRef

Description

SecretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.volumes[].scaleIO

Description

ScaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Type

`object`

Required

`gateway` `secretRef` `system`

Property	Type	Description
`fsType`	`string`	Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
`gateway`	`string`	The host address of the ScaleIO API Gateway.
`protectionDomain`	`string`	The name of the ScaleIO Protection Domain for the configured storage.
`readOnly`	`boolean`	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`	Flag to enable/disable SSL communication with Gateway, default false
`storageMode`	`string`	Indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.

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

.spec.install.spec.deployments[].spec.template.spec.volumes[].scaleIO.secretRef

Description

SecretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.volumes[].secret

Description

Secret represents a secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	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`	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`	Specify whether the Secret or its keys must be defined

Property	Type	Description
`secretName`	`string`	Name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret ↗

.spec.install.spec.deployments[].spec.template.spec.volumes[].secret.items

Description

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.install.spec.deployments[].spec.template.spec.volumes[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	The key to project.
`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`	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.install.spec.deployments[].spec.template.spec.volumes[].storageos

Description

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

Type

`object`

Property	Type	Description
`fsType`	`string`	Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4",

Property	Type	Description
		"xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
`readOnly`	`boolean`	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.install.spec.deployments[].spec.template.spec.volumes[].storageos.secretRef

Description

SecretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.install.spec.deployments[].spec.template.spec.volumes[].vsphereVolume

Description

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

Type

`object`

Required

`volumePath`

Property	Type	Description
`fsType`	`string`	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`	Storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
`storagePolicyName`	`string`	Storage Policy Based Management (SPBM) profile name.
`volumePath`	`string`	Path that identifies vSphere volume vmdk

.spec.install.spec.permissions

Type

`array`

.spec.install.spec.permissions[]

Description

StrategyDeploymentPermissions describe the rbac rules and service account needed by the install strategy

Type

`object`

Required

`rules` `serviceAccountName`

Property	Type	Description
`rules`	`array`	
`serviceAccountName`	`string`	

.spec.install.spec.permissions[].rules

Type

`array`

.spec.install.spec.permissions[].rules[]

Description

PolicyRule holds information that describes a policy rule, but does not contain information about who the rule applies to or which namespace the rule applies to.

Type

`object`

Required

`verbs`

Property	Type	Description
`apiGroups`	`array`	APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed.
`nonResourceURLs`	`array`	NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.

Property	Type	Description
`resourceNames`	`array`	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
`resources`	`array`	Resources is a list of resources this rule applies to. '*' represents all resources.
`verbs`	`array`	Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule. '*' represents all verbs.

.spec.install.spec.permissions[].rules[].apiGroups

Description

APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed.

Type

`array`

.spec.install.spec.permissions[].rules[].apiGroups[]

Type

`string`

.spec.install.spec.permissions[].rules[].nonResourceURLs

Description

NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.

Type

`array`

.spec.install.spec.permissions[].rules[].nonResourceURLs[]

Type

`string`

.spec.install.spec.permissions[].rules[].resourceNames

Description

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

Type

`array`

.spec.install.spec.permissions[].rules[].resourceNames[]

Type

`string`

.spec.install.spec.permissions[].rules[].resources

Description

Resources is a list of resources this rule applies to. '*' represents all resources.

Type

`array`

.spec.install.spec.permissions[].rules[].resources[]

Type

`string`

.spec.install.spec.permissions[].rules[].verbs

Description

Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule. '*' represents all verbs.

Type

`array`

.spec.install.spec.permissions[].rules[].verbs[]

Type

`string`

.spec.installModes

Description

InstallModes specify supported installation types

Type

`array`

.spec.installModes[]

Description

InstallMode associates an InstallModeType with a flag representing if the CSV supports it

Type

`object`

Required

`supported` `type`

Property	Type	Description
`supported`	`boolean`	
`type`	`string`	InstallModeType is a supported type of install mode for CSV installation

.spec.keywords

Type

`array`

.spec.keywords[]

Type

`string`

.spec.labels

Description

Map of string keys and values that can be used to organize and categorize (scope and select) objects.

Type

`object`

.spec.links

Type

`array`

.spec.links[]

Type

`object`

Property	Type	Description
`name`	`string`	
`url`	`string`	

.spec.maintainers

Type

`array`

.spec.maintainers[]

Type

`object`

Property	Type	Description
`email`	`string`	
`name`	`string`	

.spec.nativeAPIs

Type

`array`

.spec.nativeAPIs[]

Description

GroupVersionKind unambiguously identifies a kind. It doesn't anonymously include GroupVersion to avoid automatic coersion. It doesn't use a GroupVersion to avoid custom marshalling

Type

`object`

Required

`group` `kind` `version`

Property	Type	Description
`group`	`string`	
`kind`	`string`	
`version`	`string`	

.spec.provider

Type

`object`

Property	Type	Description
`name`	`string`	
`url`	`string`	

.spec.relatedImages

Description

List any related images, or other container images that your Operator might require to perform their functions. This list should also include operand images as well. All image references should be specified by digest (SHA) and not by tag. This field is only used during catalog creation and plays no part in cluster runtime.

Type

`array`

.spec.relatedImages[]

Type

`object`

Required

`image` `name`

Property	Type	Description
`image`	`string`	
`name`	`string`	

.spec.selector

Description

Label selector for related resources.

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.selector.matchExpressions

Description

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

Type

`array`

.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.

Property	Type	Description
`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.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.selector.matchExpressions[].values[]

Type

`string`

.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.skips

Description

The name(s) of one or more CSV(s) that should be skipped in the upgrade graph. Should match the `metadata.Name` field of the CSV that should be skipped. This field is only used during catalog creation and plays no part in cluster runtime.

Type

`array`

.spec.skips[]

Type

`string`

.spec.webhookdefinitions

Type

`array`

.spec.webhookdefinitions[]

Description

WebhookDescription provides details to OLM about required webhooks

Type

`object`

Required

`admissionReviewVersions` `generateName` `sideEffects` `type`

Property	Type	Description
`admissionReviewVersions`	`array`	
`containerPort`	`integer`	
`conversionCRDs`	`array`	
`deploymentName`	`string`	
`failurePolicy`	`string`	FailurePolicyType specifies a failure policy that defines how unrecognized errors from the admission endpoint are handled.
`generateName`	`string`	
`matchPolicy`	`string`	MatchPolicyType specifies the type of match policy.
`objectSelector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.
`reinvocationPolicy`	`string`	ReinvocationPolicyType specifies what type of policy the admission hook uses.
`rules`	`array`	

Property	Type	Description
`sideEffects`	`string`	SideEffectClass specifies the types of side effects a webhook may have.
`targetPort`	`integer|string`	
`timeoutSeconds`	`integer`	
`type`	`string`	WebhookAdmissionType is the type of admission webhooks supported by OLM
`webhookPath`	`string`	

.spec.webhookdefinitions[].admissionReviewVersions

Type

`array`

.spec.webhookdefinitions[].admissionReviewVersions[]

Type

`string`

.spec.webhookdefinitions[].conversionCRDs

Type

`array`

.spec.webhookdefinitions[].conversionCRDs[]

Type

`string`

.spec.webhookdefinitions[].objectSelector

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

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.webhookdefinitions[].objectSelector.matchExpressions

Description

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

Type

`array`

.spec.webhookdefinitions[].objectSelector.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.webhookdefinitions[].objectSelector.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.webhookdefinitions[].objectSelector.matchExpressions[].values[]

Type

`string`

.spec.webhookdefinitions[].objectSelector.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.webhookdefinitions[].rules

Type

`array`

.spec.webhookdefinitions[].rules[]

Description

RuleWithOperations is a tuple of Operations and Resources. It is recommended to make sure that all the tuple expansions are valid.

Type

`object`

Property	Type	Description
`apiGroups`	`array`	APIGroups is the API groups the resources belong to. *'' is all groups. If* '' is present, the length of the slice must be one. Required.
`apiVersions`	`array`	APIVersions is the API versions the resources belong to. *'' is all versions. If* '' is present, the length of the slice must be one. Required.
`operations`	`array`	Operations is the operations the admission hook cares about - CREATE, UPDATE, DELETE, CONNECT or * for all of those operations and any future admission operations that are added. If '*' is present, the length of the slice must be one. Required.
`resources`	`array`	Resources is a list of resources this rule applies to. For example: 'pods' means pods. 'pods/log' means the log subresource of pods. *'' means all resources, but not subresources. 'pods/*'* means all subresources of pods. *'/scale' means all scale subresources. '/*'* means all resources and their subresources. If wildcard is present, the validation rule will ensure resources do not overlap with each other. Depending on the enclosing object, subresources might not be allowed. Required.
`scope`	`string`	scope specifies the scope of this rule. Valid values are "Cluster", "Namespaced", and *"" "Cluster" means that only cluster-scoped resources will match this rule. Namespace API objects are cluster-scoped. "Namespaced" means that only namespaced resources will match this rule. ""* means

Property	Type	Description
		that there are no scope restrictions. Subresources match the scope of their parent resource. Default is "*".

.spec.webhookdefinitions[].rules[].apiGroups

Description

APIGroups is the API groups the resources belong to. '*' is all groups. If '*' is present, the length of the slice must be one. Required.

Type

`array`

.spec.webhookdefinitions[].rules[].apiGroups[]

Type

`string`

.spec.webhookdefinitions[].rules[].apiVersions

Description

APIVersions is the API versions the resources belong to. '*' is all versions. If '*' is present, the length of the slice must be one. Required.

Type

`array`

.spec.webhookdefinitions[].rules[].apiVersions[]

Type

`string`

.spec.webhookdefinitions[].rules[].operations

Description

Operations is the operations the admission hook cares about - CREATE, UPDATE, DELETE, CONNECT or * for all of those operations and any future admission operations that are added. If '*' is present, the length of the slice must be one. Required.

Type

`array`

.spec.webhookdefinitions[].rules[].operations[]

Description

OperationType specifies an operation for a request.

Type

`string`

.spec.webhookdefinitions[].rules[].resources

Description

Resources is a list of resources this rule applies to. For example: 'pods' means pods. 'pods/log' means the log subresource of pods. '*' means all resources, but not subresources. 'pods/*' means all subresources of pods. '*/scale' means all scale subresources. '*/*' means all resources and their subresources. If wildcard is present, the validation rule will ensure resources do not overlap with each other. Depending on the enclosing object, subresources might not be allowed. Required.

Type

`array`

.spec.webhookdefinitions[].rules[].resources[]

Type

`string`

.status

Description

ClusterServiceVersionStatus represents information about the status of a CSV. Status may trail the actual state of a system.

Type

`object`

Property	Type	Description
`certsLastUpdated`	`string`	Last time the owned APIService certs were updated
`certsRotateAt`	`string`	Time the owned APIService certs will rotate next
`cleanup`	`object`	CleanupStatus represents information about the status of cleanup while a CSV is pending deletion
`conditions`	`array`	List of conditions, a history of state transitions
`lastTransitionTime`	`string`	Last time the status transitioned from one status to another.
`lastUpdateTime`	`string`	Last time we updated the status
`message`	`string`	A human readable message indicating details about why the ClusterServiceVersion is in this

Property	Type	Description
		condition.
`phase`	`string`	Current condition of the ClusterServiceVersion
`reason`	`string`	A brief CamelCase message indicating details about why the ClusterServiceVersion is in this state. e.g. 'RequirementsNotMet'
`requirementStatus`	`array`	The status of each requirement for this CSV

.status.cleanup

Description

CleanupStatus represents information about the status of cleanup while a CSV is pending deletion

Type

`object`

Property	Type	Description
`pendingDeletion`	`array`	PendingDeletion is the list of custom resource objects that are pending deletion and blocked on finalizers. This indicates the progress of cleanup that is blocking CSV deletion or operator uninstall.

.status.cleanup.pendingDeletion

Description

PendingDeletion is the list of custom resource objects that are pending deletion and blocked on finalizers. This indicates the progress of cleanup that is blocking CSV deletion or operator uninstall.

Type

`array`

.status.cleanup.pendingDeletion[]

Description

ResourceList represents a list of resources which are of the same Group/Kind

Type

`object`

Required

`group` `instances` `kind`

Property	Type	Description
`group`	`string`	
`instances`	`array`	
`kind`	`string`	

.status.cleanup.pendingDeletion[].instances

Type

`array`

.status.cleanup.pendingDeletion[].instances[]

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	
`namespace`	`string`	Namespace can be empty for cluster-scoped resources

.status.conditions

Description

List of conditions, a history of state transitions

Type

`array`

.status.conditions[]

Description

Conditions appear in the status as a record of state transitions on the ClusterServiceVersion

Type

`object`

Property	Type	Description
`lastTransitionTime`	`string`	Last time the status transitioned from one status to another.
`lastUpdateTime`	`string`	Last time we updated the status

Property	Type	Description
`message`	`string`	A human readable message indicating details about why the ClusterServiceVersion is in this condition.
`phase`	`string`	Condition of the ClusterServiceVersion
`reason`	`string`	A brief CamelCase message indicating details about why the ClusterServiceVersion is in this state. e.g. 'RequirementsNotMet'

.status.requirementStatus

Description

The status of each requirement for this CSV

Type

`array`

.status.requirementStatus[]

Type

`object`

Required

`group` `kind` `message` `name` `status` `version`

Property	Type	Description
`dependents`	`array`	

Property	Type	Description
`group`	`string`	
`kind`	`string`	
`message`	`string`	
`name`	`string`	
`status`	`string`	StatusReason is a camelcased reason for the status of a RequirementStatus or DependentStatus
`uuid`	`string`	
`version`	`string`	

.status.requirementStatus[].dependents

Type

`array`

.status.requirementStatus[].dependents[]

Description

DependentStatus is the status for a dependent requirement (to prevent infinite nesting)

Type

`object`

Required

`group` `kind` `status` `version`

Property	Type	Description
`group`	`string`	

Property	Type	Description
`kind`	`string`	
`message`	`string`	
`status`	`string`	StatusReason is a camelcased reason for the status of a RequirementStatus or DependentStatus
`uuid`	`string`	
`version`	`string`	

API Endpoints

The following API endpoints are available:

- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterserviceversions`
  - `DELETE`: delete collection of ClusterServiceVersion
  - `GET`: list objects of kind ClusterServiceVersion
  - `POST`: create a new ClusterServiceVersion
- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterserviceversions/{name}`
  - `DELETE`: delete the specified ClusterServiceVersion
  - `GET`: read the specified ClusterServiceVersion
  - `PATCH`: partially update the specified ClusterServiceVersion
  - `PUT`: replace the specified ClusterServiceVersion
- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterserviceversions/{name}/status`

- `GET`: read status of the specified ClusterServiceVersion
- `PATCH`: partially update status of the specified ClusterServiceVersion
- `PUT`: replace status of the specified ClusterServiceVersion

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterserviceversions

HTTP method

`DELETE`

Description

delete collection of ClusterServiceVersion

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind ClusterServiceVersion

HTTP responses

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

HTTP method

`POST`

Description

create a new ClusterServiceVersion

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`	`ClusterServiceVersion` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterserviceversions/{name}

HTTP method

`DELETE`

Description

delete the specified ClusterServiceVersion

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 ClusterServiceVersion

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ClusterServiceVersion

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	`ClusterServiceVersion` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ClusterServiceVersion

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`	`ClusterServiceVersion` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterserviceversions/{name}/status

HTTP method

`GET`

Description

read status of the specified ClusterServiceVersion

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ClusterServiceVersion

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	`ClusterServiceVersion` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ClusterServiceVersion

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`	`ClusterServiceVersion` schema	`application/json` formatted

HTTP responses

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

InstallPlan [operators.coreos.com/v1alpha1]

Description

InstallPlan defines the installation of a set of operators.

Type

`object`

Required

`metadata` `spec`

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`	InstallPlanSpec defines a set of Application resources to be installed
`status`	`object`	InstallPlanStatus represents the information about the status of steps required to complete installation. Status may trail the actual state of a system.

.spec

Description

InstallPlanSpec defines a set of Application resources to be installed

Type

`object`

Required

`approval` `approved` `clusterServiceVersionNames`

Property	Type	Description
`approval`	`string`	Approval is the user approval policy for an InstallPlan. It must be one of "Automatic" or "Manual".
`approved`	`boolean`	
`clusterServiceVersionNames`	`array`	
`generation`	`integer`	
`source`	`string`	
`sourceNamespace`	`string`	

.spec.clusterServiceVersionNames

Type

`array`

.spec.clusterServiceVersionNames[]

Type

`string`

.status

Description

InstallPlanStatus represents the information about the status of steps required to complete installation. Status may trail the actual state of a system.

Type

`object`

Required

`catalogSources` `phase`

Property	Type	Description
`attenuatedServiceAccountRef`	`object`	AttenuatedServiceAccountRef references the service account that is used to do scoped operator install.
`bundleLookups`	`array`	BundleLookups is the set of in-progress requests to pull and unpackage bundle content to the cluster.
`catalogSources`	`array`	
`conditions`	`array`	
`message`	`string`	Message is a human-readable message containing detailed information that may be important to understanding why the plan has its current status.
`phase`	`string`	InstallPlanPhase is the current status of a InstallPlan as a whole.
`plan`	`array`	
`startTime`	`string`	StartTime is the time when the controller began applying the resources listed in the plan to the cluster.

.status.attenuatedServiceAccountRef

Description

AttenuatedServiceAccountRef references the service account that is used to do scoped operator install.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.status.bundleLookups

Description

BundleLookups is the set of in-progress requests to pull and unpackage bundle content to the cluster.

Type

`array`

.status.bundleLookups[]

Description

BundleLookup is a request to pull and unpackage the content of a bundle to the cluster.

Type

`object`

Required

`catalogSourceRef` `identifier` `path` `replaces`

Property	Type	Description
`catalogSourceRef`	`object`	CatalogSourceRef is a reference to the CatalogSource the bundle path was resolved from.
`conditions`	`array`	Conditions represents the overall state of a BundleLookup.
`identifier`	`string`	Identifier is the catalog-unique name of the operator (the name of the CSV for bundles that contain CSVs)
`path`	`string`	Path refers to the location of a bundle to pull. It's typically an image reference.
`properties`	`string`	The effective properties of the unpacked bundle.
`replaces`	`string`	Replaces is the name of the bundle to replace with the one found at Path.

.status.bundleLookups[].catalogSourceRef

Description

CatalogSourceRef is a reference to the CatalogSource the bundle path was resolved from.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-

Property	Type	Description
		with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.status.bundleLookups[].conditions

Description

Conditions represents the overall state of a BundleLookup.

Type

`array`

.status.bundleLookups[].conditions[]

Type

`object`

Required

`status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	Last time the condition transitioned from one status to another.
`lastUpdateTime`	`string`	Last time the condition was probed.
`message`	`string`	A human readable message indicating details about the transition.
`reason`	`string`	The reason for the condition's last transition.
`status`	`string`	Status of the condition, one of True, False, Unknown.
`type`	`string`	Type of condition.

.status.catalogSources

Type

`array`

.status.catalogSources[]

Type

`string`

.status.conditions

Type

`array`

.status.conditions[]

Description

InstallPlanCondition represents the overall status of the execution of an InstallPlan.

Type

`object`

Property	Type	Description
`lastTransitionTime`	`string`	
`lastUpdateTime`	`string`	
`message`	`string`	
`reason`	`string`	ConditionReason is a camelcased reason for the state transition.
`status`	`string`	
`type`	`string`	InstallPlanConditionType describes the state of an InstallPlan at a certain point as a whole.

.status.plan

Type

`array`

.status.plan[]

Description

Step represents the status of an individual step in an InstallPlan.

Type

`object`

Required

`resolving` `resource` `status`

Property	Type	Description
`resolving`	`string`	
`resource`	`object`	StepResource represents the status of a resource to be tracked by an InstallPlan.
`status`	`string`	StepStatus is the current status of a particular resource an in InstallPlan

.status.plan[].resource

Description

StepResource represents the status of a resource to be tracked by an InstallPlan.

Type

`object`

Required

`group` `kind` `name` `sourceName` `sourceNamespace` `version`

Property	Type	Description
`group`	`string`	
`kind`	`string`	
`manifest`	`string`	
`name`	`string`	
`sourceName`	`string`	
`sourceNamespace`	`string`	
`version`	`string`	

API Endpoints

The following API endpoints are available:

- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans`
  - `DELETE`: delete collection of InstallPlan
  - `GET`: list objects of kind InstallPlan
  - `POST`: create a new InstallPlan
- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans/{name}`
  - `DELETE`: delete the specified InstallPlan
  - `GET`: read the specified InstallPlan
  - `PATCH`: partially update the specified InstallPlan
  - `PUT`: replace the specified InstallPlan
- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans/{name}/status`

- `GET`: read status of the specified InstallPlan
- `PATCH`: partially update status of the specified InstallPlan
- `PUT`: replace status of the specified InstallPlan

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans

HTTP method

`DELETE`

Description

delete collection of InstallPlan

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind InstallPlan

HTTP responses

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

HTTP method

`POST`

Description

create a new InstallPlan

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`	`InstallPlan` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans/{name}

HTTP method

`DELETE`

Description

delete the specified InstallPlan

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 InstallPlan

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified InstallPlan

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	`InstallPlan` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified InstallPlan

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`	`InstallPlan` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans/{name}/status

HTTP method

`GET`

Description

read status of the specified InstallPlan

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified InstallPlan

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	`InstallPlan` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified InstallPlan

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`	`InstallPlan` schema	`application/json` formatted

HTTP responses

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

OLMConfig [operators.coreos.com/v1]

Description

OLMConfig is a resource responsible for configuring OLM.

Type

`object`

Required

`metadata`

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 ↗
`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

Property	Type	Description
		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`	OLMConfigSpec is the spec for an OLMConfig resource.
`status`	`object`	OLMConfigStatus is the status for an OLMConfig resource.

.spec

Description

OLMConfigSpec is the spec for an OLMConfig resource.

Type

`object`

Property	Type	Description
`features`	`object`	Features contains the list of configurable OLM features.

.spec.features

Description

Features contains the list of configurable OLM features.

Type

`object`

Property	Type	Description
`disableCopiedCSVs`	`boolean`	DisableCopiedCSVs is used to disable OLM's "Copied CSV" feature for operators installed at the cluster scope, where a cluster scoped operator is one that has been installed in an OperatorGroup that targets all namespaces. When reenabled, OLM will recreate the "Copied CSVs" for each cluster scoped operator.

.status

Description

OLMConfigStatus is the status for an OLMConfig resource.

Type

`object`

Property	Type	Description
`conditions`	`array`	

.status.conditions

Type

`array`

.status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"` // other fields }

Type

`object`

Required

`lastTransitionTime` `message` `reason` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.

Property	Type	Description
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. --- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

API Endpoints

The following API endpoints are available:

- `/apis/operators.coreos.com/v1/olmconfigs`
  - `DELETE`: delete collection of OLMConfig
  - `GET`: list objects of kind OLMConfig
  - `POST`: create a new OLMConfig

- `/apis/operators.coreos.com/v1/olmconfigs/{name}`
  - `DELETE`: delete the specified OLMConfig
  - `GET`: read the specified OLMConfig
  - `PATCH`: partially update the specified OLMConfig
  - `PUT`: replace the specified OLMConfig
- `/apis/operators.coreos.com/v1/olmconfigs/{name}/status`
  - `GET`: read status of the specified OLMConfig
  - `PATCH`: partially update status of the specified OLMConfig
  - `PUT`: replace status of the specified OLMConfig

/apis/operators.coreos.com/v1/olmconfigs

HTTP method

`DELETE`

Description

delete collection of OLMConfig

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind OLMConfig

HTTP responses

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

HTTP method

`POST`

Description

create a new OLMConfig

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`	`OLMConfig` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1/olmconfigs/{name}

HTTP method

`DELETE`

Description

delete the specified OLMConfig

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 OLMConfig

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified OLMConfig

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:

Parameter	Type	Description
		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	`OLMConfig` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified OLMConfig

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`	`OLMConfig` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1/olmconfigs/{name}/status

HTTP method

`GET`

Description

read status of the specified OLMConfig

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified OLMConfig

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

Parameter	Type	Description
		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	`OLMConfig` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified OLMConfig

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

Parameter	Type	Description
		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`	`OLMConfig` schema	`application/json` formatted

HTTP responses

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

OperatorCondition [operators.coreos.com/v1]

Description

OperatorCondition is a Custom Resource of type `OperatorCondition` which is used to convey information to OLM about the state of an operator.

Type

`object`

Required

`metadata`

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`	OperatorConditionSpec allows an operator to report state to OLM and provides cluster admin with the ability to manually override state reported by the operator.
`status`	`object`	OperatorConditionStatus allows OLM to convey which conditions have been observed.

.spec

Description

OperatorConditionSpec allows an operator to report state to OLM and provides cluster admin with the ability to manually override state reported by the operator.

Type

`object`

Property	Type	Description
`conditions`	`array`	
`deployments`	`array`	
`overrides`	`array`	
`serviceAccounts`	`array`	

.spec.conditions

Type

`array`

.spec.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"` // other fields }

Type

`object`

Required

`lastTransitionTime` `message` `reason` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This

Property	Type	Description
		should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. --- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

.spec.deployments

Type

`array`

.spec.deployments[]

Type

`string`

.spec.overrides

Type

`array`

.spec.overrides[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known

.status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"` // other fields }

Type

`object`

Required

`message` `reason` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last

Property	Type	Description
		transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. --- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

.spec.serviceAccounts

Type

`array`

.spec.serviceAccounts[]

Type

`string`

.status

Description

OperatorConditionStatus allows OLM to convey which conditions have been observed.

Type

`object`

Property	Type	Description
`conditions`	`array`	

.status.conditions

Type

`array`

.status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"` // other fields }

Type

`object`

Required

`lastTransitionTime` `message` `reason` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This

Property	Type	Description
		should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. --- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

API Endpoints

The following API endpoints are available:

- `/apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions`
  - `DELETE`: delete collection of OperatorCondition
  - `GET`: list objects of kind OperatorCondition
  - `POST`: create a new OperatorCondition
- `/apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions/{name}`
  - `DELETE`: delete the specified OperatorCondition
  - `GET`: read the specified OperatorCondition
  - `PATCH`: partially update the specified OperatorCondition
  - `PUT`: replace the specified OperatorCondition
- `/apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions/{name}/status`
  - `GET`: read status of the specified OperatorCondition
  - `PATCH`: partially update status of the specified OperatorCondition

- `PUT`: replace status of the specified OperatorCondition

/apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions

HTTP method

`DELETE`

Description

delete collection of OperatorCondition

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind OperatorCondition

HTTP responses

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

HTTP method

`POST`

Description

create a new OperatorCondition

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`	`OperatorCondition` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`OperatorCondition` schema

HTTP code	Response body
201 - Created	`OperatorCondition` schema
202 - Accepted	`OperatorCondition` schema
401 - Unauthorized	Empty

/apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions/{name}

HTTP method

`DELETE`

Description

delete the specified OperatorCondition

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 OperatorCondition

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified OperatorCondition

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

Parameter	Type	Description
		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	`OperatorCondition` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified OperatorCondition

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

Parameter	Type	Description
		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`	`OperatorCondition` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions/{name}/status

HTTP method

`GET`

Description

read status of the specified OperatorCondition

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified OperatorCondition

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	`OperatorCondition` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified OperatorCondition

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`	`OperatorCondition` schema	`application/json` formatted

HTTP responses

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

OperatorGroup [operators.coreos.com/v1]

Description

OperatorGroup is the unit of multitenancy for OLM managed operators. It constrains the installation of operators in its namespace to a specified set of target namespaces.

Type

`object`

Required

`metadata`

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 ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint

Property	Type	Description
		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`	OperatorGroupSpec is the spec for an OperatorGroup resource.
`status`	`object`	OperatorGroupStatus is the status for an OperatorGroupResource.

.spec

Description

OperatorGroupSpec is the spec for an OperatorGroup resource.

Type

`object`

Property	Type	Description
`selector`	`object`	Selector selects the OperatorGroup's target namespaces.

Property	Type	Description
`serviceAccountName`	`string`	ServiceAccountName is the admin specified service account which will be used to deploy operator(s) in this operator group.
`staticProvidedAPIs`	`boolean`	Static tells OLM not to update the OperatorGroup's providedAPIs annotation
`targetNamespaces`	`array`	TargetNamespaces is an explicit set of namespaces to target. If it is set, Selector is ignored.

.spec.selector

Description

Selector selects the OperatorGroup's target 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

Property	Type	Description
		"key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.selector.matchExpressions

Description

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

Type

`array`

.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.

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.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.selector.matchExpressions[].values[]

Type

`string`

.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.targetNamespaces

Description

TargetNamespaces is an explicit set of namespaces to target. If it is set, Selector is ignored.

Type

`array`

.spec.targetNamespaces[]

Type

`string`

.status

Description

OperatorGroupStatus is the status for an OperatorGroupResource.

Type

`object`

Required

`lastUpdated`

Property	Type	Description
`conditions`	`array`	Conditions is an array of the OperatorGroup's conditions.
`lastUpdated`	`string`	LastUpdated is a timestamp of the last time the OperatorGroup's status was Updated.
`namespaces`	`array`	Namespaces is the set of target namespaces for the OperatorGroup.

Property	Type	Description
`serviceAccountRef`	`object`	ServiceAccountRef references the service account object specified.

.status.conditions

Description

Conditions is an array of the OperatorGroup's conditions.

Type

`array`

.status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"` // other fields }

Type

`object`

Required

`lastTransitionTime` `message` `reason` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This

Property	Type	Description
		should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. --- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

.status.namespaces

Description

Namespaces is the set of target namespaces for the OperatorGroup.

Type

`array`

.status.namespaces[]

Type

`string`

.status.serviceAccountRef

Description

ServiceAccountRef references the service account object specified.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗

Property	Type	Description
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

API Endpoints

The following API endpoints are available:

- `/apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups`
  - `DELETE`: delete collection of OperatorGroup
  - `GET`: list objects of kind OperatorGroup
  - `POST`: create a new OperatorGroup
- `/apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups/{name}`
  - `DELETE`: delete the specified OperatorGroup
  - `GET`: read the specified OperatorGroup
  - `PATCH`: partially update the specified OperatorGroup
  - `PUT`: replace the specified OperatorGroup
- `/apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups/{name}/status`
  - `GET`: read status of the specified OperatorGroup

- `PATCH`: partially update status of the specified OperatorGroup
- `PUT`: replace status of the specified OperatorGroup

/apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups

HTTP method

`DELETE`

Description

delete collection of OperatorGroup

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind OperatorGroup

HTTP responses

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

HTTP method

`POST`

Description

create a new OperatorGroup

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`	`OperatorGroup` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`OperatorGroup` schema

HTTP code	Response body
201 - Created	`OperatorGroup` schema
202 - Accepted	`OperatorGroup` schema
401 - Unauthorized	Empty

/apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups/{name}

HTTP method

`DELETE`

Description

delete the specified OperatorGroup

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 OperatorGroup

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified OperatorGroup

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

Parameter	Type	Description
		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	`OperatorGroup` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified OperatorGroup

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

Parameter	Type	Description
		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`	`OperatorGroup` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups/{name}/status

HTTP method

`GET`

Description

read status of the specified OperatorGroup

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified OperatorGroup

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	`OperatorGroup` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified OperatorGroup

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`	`OperatorGroup` schema	`application/json` formatted

HTTP responses

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

Subscription [operators.coreos.com/v1alpha1]

Description

Subscription keeps operators up to date by tracking changes to Catalogs.

Type

`object`

Required

`metadata` `spec`

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`	SubscriptionSpec defines an Application that can be installed
`status`	`object`	

.spec

Description

SubscriptionSpec defines an Application that can be installed

Type

`object`

Required

`name` `source` `sourceNamespace`

Property	Type	Description
`channel`	`string`	

Property	Type	Description
`config`	`object`	SubscriptionConfig contains configuration specified for a subscription.
`installPlanApproval`	`string`	Approval is the user approval policy for an InstallPlan. It must be one of "Automatic" or "Manual".
`name`	`string`	
`source`	`string`	
`sourceNamespace`	`string`	
`startingCSV`	`string`	

.spec.config

Description

SubscriptionConfig contains configuration specified for a subscription.

Type

`object`

Property	Type	Description
`env`	`array`	Env is a list of environment variables to set in the container. Cannot be updated.
`envFrom`	`array`	EnvFrom is a list of sources to populate environment variables in the container. The keys defined within a

Property	Type	Description
		source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Immutable.
`nodeSelector`	`object`	NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/ ↗
`resources`	`object`	Resources represents compute resources required by this container. Immutable. More info: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/ ↗
`selector`	`object`	Selector is the label selector for pods to be configured. Existing ReplicaSets whose pods are selected by this will be the ones affected by this deployment. It must match the pod template's labels.
`tolerations`	`array`	Tolerations are the pod's tolerations.
`volumeMounts`	`array`	List of VolumeMounts to set in the container.

Property	Type	Description
`volumes`	`array`	List of Volumes to set in the podSpec.

.spec.config.env

Description

Env is a list of environment variables to set in the container. Cannot be updated.

Type

`array`

.spec.config.env[]

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

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

Property	Type	Description
		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.config.env[].valueFrom

Description

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

Type

`object`

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

Property	Type	Description
`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.config.env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?

Property	Type	Description
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.config.env[].valueFrom.fieldRef

Description

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

Type

`object`

Required

`fieldPath`

Property	Type	Description
`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.config.env[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.config.env[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key of the secret to select from. Must be a valid secret key.

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.config.envFrom

Description

EnvFrom is a list of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Immutable.

Type

`array`

.spec.config.envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

`object`

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

.spec.config.envFrom[].configMapRef

Description

The ConfigMap to select from

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap must be defined

.spec.config.envFrom[].secretRef

Description

The Secret to select from

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret must be defined

.spec.config.nodeSelector

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/

Type

`object`

.spec.config.resources

Description

Resources represents compute resources required by this container. Immutable. More info: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/

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. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.config.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.config.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. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Type

`object`

.spec.config.selector

Description

Selector is the label selector for pods to be configured. Existing ReplicaSets whose pods are selected by this will be the ones affected by this deployment. It must match the pod template's labels.

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.config.selector.matchExpressions

Description

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

Type

`array`

.spec.config.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.config.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.config.selector.matchExpressions[].values[]

Type

`string`

.spec.config.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.config.tolerations

Description

Tolerations are the pod's tolerations.

Type

`array`

.spec.config.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.config.volumeMounts

Description

List of VolumeMounts to set in the container.

Type

`array`

.spec.config.volumeMounts[]

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.
`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.

Property	Type	Description
`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.config.volumes

Description

List of Volumes to set in the podSpec.

Type

`array`

.spec.config.volumes[]

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

`object`

Required

`name`

Property	Type	Description
`awsElasticBlockStore`	`object`	AWSElasticBlockStore represents an AWS Disk resc to a kubelet's host machine and then exposed to the https://kubernetes.io/docs/concepts/storage/volumes ↗
`azureDisk`	`object`	AzureDisk represents an Azure Data Disk mount on mount to the pod.
`azureFile`	`object`	AzureFile represents an Azure File Service mount or mount to the pod.
`cephfs`	`object`	CephFS represents a Ceph FS mount on the host th lifetime
`cinder`	`object`	Cinder represents a cinder volume attached and mo host machine. More info: https://examples.k8s.io/my pd/README.md ↗
`configMap`	`object`	ConfigMap represents a configMap that should popu
`csi`	`object`	CSI (Container Storage Interface) represents ephem handled by certain external CSI drivers (Beta feature

Property	Type	Description
`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 the volume is only needed while the pod runs, b) fea volumes like restoring from snapshot or capacity trac the storage driver is specified through a storage clas driver supports dynamic volume provisioning throug PersistentVolumeClaim (see EphemeralVolumeSour information on the connection between this volume t PersistentVolumeClaim). Use PersistentVolumeClai vendor-specific APIs for volumes that persist for long an individual pod. Use CSI for light-weight local eph CSI driver is meant to be used that way - see the do driver for more information. A pod can use both type volumes and persistent volumes at the same time. T and only available when the GenericEphemeralVolu enabled.
`fc`	`object`	FC represents a Fibre Channel resource that is attac host machine and then exposed to the pod.

Property	Type	Description
`flexVolume`	`object`	FlexVolume represents a generic volume resource tl provisioned/attached using an exec based plugin.
`flocker`	`object`	Flocker represents a Flocker volume attached to a k 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
`gitRepo`	`object`	GitRepo represents a git repository at a particular re DEPRECATED: GitRepo is deprecated. To provision repo, mount an EmptyDir into an InitContainer that c git, then mount the EmptyDir into the Pod's containe
`glusterfs`	`object`	Glusterfs represents a Glusterfs mount on the host t lifetime. More info: https://examples.k8s.io/volumes/ ↗
`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 TODO(jonesdl) We need to restrict who can use hos and who can/can not mount host directories as read

Property	Type	Description
`iscsi`	`object`	ISCSI represents an ISCSI Disk resource that is atta host machine and then exposed to the pod. More inf https://examples.k8s.io/volumes/iscsi/README.md ↗
`name`	`string`	Volume's name. Must be a DNS_LABEL and unique info: https://kubernetes.io/docs/concepts/overview/w objects/names/#names ↗
`nfs`	`object`	NFS represents an NFS mount on the host that shar More info: https://kubernetes.io/docs/concepts/stora
`persistentVolumeClaim`	`object`	PersistentVolumeClaimVolumeSource represents a PersistentVolumeClaim in the same namespace. Mo https://kubernetes.io/docs/concepts/storage/persiste volumes#persistentvolumeclaims ↗
`photonPersistentDisk`	`object`	PhotonPersistentDisk represents a PhotonControlle attached and mounted on kubelets host machine
`portworxVolume`	`object`	PortworxVolume represents a portworx volume attac kubelets host machine
`projected`	`object`	Items for all in one resources secrets, configmaps, a

Property	Type	Description
`quobyte`	`object`	Quobyte represents a Quobyte mount on the host th 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
`storageos`	`object`	StorageOS represents a StorageOS volume attache Kubernetes nodes.
`vsphereVolume`	`object`	VsphereVolume represents a vSphere volume attacl kubelets host machine

.spec.config.volumes[].awsElasticBlockStore

Description

AWSElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	Filesystem type of the volume that you want to mount. Tip: Ensu filesystem type is supported by the host operating system. Exan "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗ TODO: how do we prevent errors in the filesystem from comp the machine
`partition`	`integer`	The partition in the volume that you want to mount. If omitted, th is to mount by volume name. Examples: For volume /dev/sda1, specify the partition as "1". Similarly, the volume partition for /de "0" (or you can leave the property empty).
`readOnly`	`boolean`	Specify "true" to force and set the ReadOnly property in Volume "true". If omitted, the default is "false". More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗
`volumeID`	`string`	Unique ID of the persistent disk resource in AWS (Amazon EBS More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗

.spec.config.volumes[].azureDisk

Description

AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

`object`

Required

`diskName` `diskURI`

Property	Type	Description
`cachingMode`	`string`	Host Caching mode: None, Read Only, Read Write.
`diskName`	`string`	The Name of the data disk in the blob storage
`diskURI`	`string`	The URI the data disk in the blob storage
`fsType`	`string`	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`	Expected values Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

Property	Type	Description
`readOnly`	`boolean`	Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

.spec.config.volumes[].azureFile

Description

AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

Type

`object`

Required

`secretName` `shareName`

Property	Type	Description
`readOnly`	`boolean`	Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretName`	`string`	the name of secret that contains Azure Storage Account Name and Key
`shareName`	`string`	Share Name

.spec.config.volumes[].cephfs

Description

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

Type

`object`

Required

`monitors`

Property	Type	Description
`monitors`	`array`	Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`path`	`string`	Optional: Used as the mounted root, rather than the full Ceph tree, default is /
`readOnly`	`boolean`	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`	Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

Property	Type	Description
`secretRef`	`object`	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`	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.config.volumes[].cephfs.monitors

Description

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.config.volumes[].cephfs.monitors[]

Type

`string`

.spec.config.volumes[].cephfs.secretRef

Description

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. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.config.volumes[].cinder

Description

Cinder represents a cinder volume attached and mounted on kubelets host machine. More info: https://examples.k8s.io/mysql-cinder-pd/README.md

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	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`	Optional: 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 ↗

Property	Type	Description
`secretRef`	`object`	Optional: points to a secret object containing parameters used to connect to OpenStack.
`volumeID`	`string`	volume id used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

.spec.config.volumes[].cinder.secretRef

Description

Optional: points to a secret object containing parameters used to connect to OpenStack.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.config.volumes[].configMap

Description

ConfigMap represents a configMap that should populate this volume

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	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`	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. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap or its keys must be defined

.spec.config.volumes[].configMap.items

Description

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.config.volumes[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	The key to project.
`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.

Property	Type	Description
`path`	`string`	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.config.volumes[].csi

Description

CSI (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Type

`object`

Required

`driver`

Property	Type	Description
`driver`	`string`	Driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
`fsType`	`string`	Filesystem type 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

Property	Type	Description
		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`	Specifies a read-only configuration for the volume. Defaults to false (read/write).
`volumeAttributes`	`object`	VolumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

.spec.config.volumes[].csi.nodePublishSecretRef

Description

NodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?

.spec.config.volumes[].csi.volumeAttributes

Description

VolumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

`object`

.spec.config.volumes[].downwardAPI

Description

DownwardAPI represents downward API about the pod that should populate this volume

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the

Property	Type	Description
		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.config.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

`array`

.spec.config.volumes[].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 and namespace 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.config.volumes[].downwardAPI.items[].fieldRef

Description

Required: Selects a field of the pod: only annotations, labels, name and namespace 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.config.volumes[].downwardAPI.items[].resourceFieldRef

Description

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

Type

`object`

Required

`resource`

Property	Type	Description
`containerName`	`string`	Container name: required for volumes, optional for env vars
`divisor`	`integer|string`	Specifies the output format of the exposed resources, defaults to "1"

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

.spec.config.volumes[].emptyDir

Description

EmptyDir represents a temporary directory that shares a pod's lifetime. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Type

`object`

Property	Type	Description
`medium`	`string`	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#empty ↗
`sizeLimit`	`integer|string`	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 b 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: http://kubernetes.io/docs/user-guide/volumes#emptydir

.spec.config.volumes[].ephemeral

Description

Ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual pod. Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time. This is a beta feature and only available when the GenericEphemeralVolume feature gate is enabled.

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

Property	Type	Description
		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.config.volumes[].ephemeral.volumeClaimTemplate

Description

Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be `<pod name>-<volume name>` where `<volume name>` is the name from the `PodSpec.Volumes` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

Type

`object`

Required

`spec`

Property	Type	Description
`metadata`	`ObjectMeta`	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.
`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.config.volumes[].ephemeral.volumeClaimTemplate.spec

Description

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

Type

`object`

Property	Type	Description
`accessModes`	`array`	AccessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1 ↗
`dataSource`	`object`	This field can be used to specify either: * An existing VolumeSnapshot object

Property	Type	Description
		(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. If the AnyVolumeDataSource feature gate is enabled, this field will always have the same contents as the DataSourceRef field.
`dataSourceRef`	`object`	Specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any local 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, 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. There are two 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. (Alpha) Using this field requires the AnyVolumeDataSource feature gate to be enabled.

Property	Type	Description
`resources`	`object`	Resources represents the minimum resources the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources ↗
`selector`	`object`	A label query over volumes to consider for binding.
`storageClassName`	`string`	Name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1 ↗
`volumeMode`	`string`	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.
`volumeName`	`string`	VolumeName is the binding reference to the PersistentVolume backing this claim.

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes

Description

AccessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1

Type

`array`

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]

Type

`string`

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

Description

This 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. If the AnyVolumeDataSource feature gate is enabled, this field will always have the same contents as the DataSourceRef field.

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

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef

Description

Specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any local 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, 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. There are two 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. (Alpha) Using this field requires the AnyVolumeDataSource feature gate to be enabled.

Type

`object`

Required

`kind` `name`

Property	Type	Description
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

Resources represents the minimum resources the volume should have. 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

Property	Type	Description
		defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits

Description

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

Type

`object`

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Type

`object`

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector

Description

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.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions

Description

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

Type

`array`

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values

Description

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

Type

`array`

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]

Type

`string`

.spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels

Description

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

Type

`object`

.spec.config.volumes[].fc

Description

FC represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Type

`object`

Property	Type	Description
`fsType`	`string`	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. TODO: how do we prevent errors in the filesystem from compromising the machine
`lun`	`integer`	Optional: FC target lun number

Property	Type	Description
`readOnly`	`boolean`	Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`targetWWNs`	`array`	Optional: FC target worldwide names (WWNs)
`wwids`	`array`	Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

.spec.config.volumes[].fc.targetWWNs

Description

Optional: FC target worldwide names (WWNs)

Type

`array`

.spec.config.volumes[].fc.targetWWNs[]

Type

`string`

.spec.config.volumes[].fc.wwids

Description

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.config.volumes[].fc.wwids[]

Type

`string`

.spec.config.volumes[].flexVolume

Description

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

Type

`object`

Required

`driver`

Property	Type	Description
`driver`	`string`	Driver is the name of the driver to use for this volume.
`fsType`	`string`	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`	Optional: Extra command options if any.

Property	Type	Description
`readOnly`	`boolean`	Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretRef`	`object`	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.config.volumes[].flexVolume.options

Description

Optional: Extra command options if any.

Type

`object`

.spec.config.volumes[].flexVolume.secretRef

Description

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. More info: https://kubernetes.io/docs/concepts/overview/working-with-

Property	Type	Description
		objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.config.volumes[].flocker

Description

Flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Type

`object`

Property	Type	Description
`datasetName`	`string`	Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
`datasetUUID`	`string`	UUID of the dataset. This is unique identifier of a Flocker dataset

.spec.config.volumes[].gcePersistentDisk

Description

GCEPersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Type

`object`

Required

`pdName`

Property	Type	Description
`fsType`	`string`	Filesystem type of the volume that you want to mount. Tip: Ensu 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#gcepersis ↗ TODO: how do we prevent errors in the filesystem from compromising the machine
`partition`	`integer`	The partition in the volume that you want to mount. If omitted, th default is to mount by volume name. Examples: For volume /dev you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗
`pdName`	`string`	Unique name of the PD resource in GCE. Used to identify the di 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.config.volumes[].gitRepo

Description

GitRepo represents a git repository at a particular revision. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
`directory`	`string`	Target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
`repository`	`string`	Repository URL
`revision`	`string`	Commit hash for the specified revision.

.spec.config.volumes[].glusterfs

Description

Glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/glusterfs/README.md

Type

`object`

Required

`endpoints` `path`

Property	Type	Description
`endpoints`	`string`	EndpointsName 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-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.config.volumes[].hostPath

Description

HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath --- TODO(jonesdl) We need to restrict who can use host directory mounts and who can/can not mount host directories as read/write.

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

.spec.config.volumes[].iscsi

Description

ISCSI represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://examples.k8s.io/volumes/iscsi/README.md

Type

`object`

Required

`iqn` `lun` `targetPortal`

Property	Type	Description
`chapAuthDiscovery`	`boolean`	whether support iSCSI Discovery CHAP authentication
`chapAuthSession`	`boolean`	whether support iSCSI Session CHAP authentication

Property	Type	Description
`fsType`	`string`	Filesystem type of the volume that you want to mount. Ti Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitl inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#isc ↗ TODO: how do we prevent errors in the filesystem fror compromising the machine
`initiatorName`	`string`	Custom iSCSI Initiator Name. If initiatorName is specifiec with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
`iqn`	`string`	Target iSCSI Qualified Name.
`iscsiInterface`	`string`	iSCSI Interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
`lun`	`integer`	iSCSI Target Lun number.
`portals`	`array`	iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCF ports 860 and 3260).
`readOnly`	`boolean`	ReadOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.

Property	Type	Description
`secretRef`	`object`	CHAP Secret for iSCSI target and initiator authentication
`targetPortal`	`string`	iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

.spec.config.volumes[].iscsi.portals

Description

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.config.volumes[].iscsi.portals[]

Type

`string`

.spec.config.volumes[].iscsi.secretRef

Description

CHAP Secret for iSCSI target and initiator authentication

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?

.spec.config.volumes[].nfs

Description

NFS represents an NFS mount on the host that shares a pod's lifetime More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

Type

`object`

Required

`path` `server`

Property	Type	Description
`path`	`string`	Path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
`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

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

.spec.config.volumes[].persistentVolumeClaim

Description

PersistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Type

`object`

Required

`claimName`

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

.spec.config.volumes[].photonPersistentDisk

Description

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

Type

`object`

Required

`pdID`

Property	Type	Description
`fsType`	`string`	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`	ID that identifies Photon Controller persistent disk

.spec.config.volumes[].portworxVolume

Description

PortworxVolume represents a portworx volume attached and mounted on kubelets host machine

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	FSType represents the filesystem type to mount Must be a filesystem type supported by the host operating system.

Property	Type	Description
		Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
`readOnly`	`boolean`	Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`volumeID`	`string`	VolumeID uniquely identifies a Portworx volume

.spec.config.volumes[].projected

Description

Items for all in one resources secrets, configmaps, and downward API

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	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`	list of volume projections

.spec.config.volumes[].projected.sources

Description

list of volume projections

Type

`array`

.spec.config.volumes[].projected.sources[]

Description

Projection that may be projected along with other supported volume types

Type

`object`

Property	Type	Description
`configMap`	`object`	information about the configMap data to project
`downwardAPI`	`object`	information about the downwardAPI data to project
`secret`	`object`	information about the secret data to project
`serviceAccountToken`	`object`	information about the serviceAccountToken data to project

.spec.config.volumes[].projected.sources[].configMap

Description

information about the configMap data to project

Type

`object`

Property	Type	Description
`items`	`array`	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. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the ConfigMap or its keys must be defined

.spec.config.volumes[].projected.sources[].configMap.items

Description

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.config.volumes[].projected.sources[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	The key to project.
`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.

Property	Type	Description
`path`	`string`	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.config.volumes[].projected.sources[].downwardAPI

Description

information about the downwardAPI data to project

Type

`object`

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

.spec.config.volumes[].projected.sources[].downwardAPI.items

Description

Items is a list of DownwardAPIVolume file

Type

`array`

.spec.config.volumes[].projected.sources[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

`object`

Required

`path`

Property	Type	Description
`fieldRef`	`object`	Required: Selects a field of the pod: only annotations, labels, name and namespace are supported.
`mode`	`integer`	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`path`	`string`	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'

Property	Type	Description
`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.config.volumes[].projected.sources[].downwardAPI.items[].fieldRef

Description

Required: Selects a field of the pod: only annotations, labels, name and namespace 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.config.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.config.volumes[].projected.sources[].secret

Description

information about the secret data to project

Type

`object`

Property	Type	Description
`items`	`array`	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

Property	Type	Description
		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. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.config.volumes[].projected.sources[].secret.items

Description

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.config.volumes[].projected.sources[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	The key to project.
`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`	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.config.volumes[].projected.sources[].serviceAccountToken

Description

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.config.volumes[].quobyte

Description

Quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Type

`object`

Required

`registry` `volume`

Property	Type	Description
`group`	`string`	Group to map volume access to Default is no group
`readOnly`	`boolean`	ReadOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
`registry`	`string`	Registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
`tenant`	`string`	Tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
`user`	`string`	User to map volume access to Defaults to serivceaccount user
`volume`	`string`	Volume is a string that references an already created Quobyte volume by name.

.spec.config.volumes[].rbd

Description

RBD represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/rbd/README.md

Type

`object`

Required

`image` `monitors`

Property	Type	Description
`fsType`	`string`	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 ↗ TODO: how do we prevent errors in the filesystem from compromising the machine
`image`	`string`	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`	A collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

Property	Type	Description
`pool`	`string`	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`	The rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

.spec.config.volumes[].rbd.monitors

Description

A collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Type

`array`

.spec.config.volumes[].rbd.monitors[]

Type

`string`

.spec.config.volumes[].rbd.secretRef

Description

SecretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.config.volumes[].scaleIO

Description

ScaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Type

`object`

Required

`gateway` `secretRef` `system`

Property	Type	Description
`fsType`	`string`	Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
`gateway`	`string`	The host address of the ScaleIO API Gateway.
`protectionDomain`	`string`	The name of the ScaleIO Protection Domain for the configured storage.
`readOnly`	`boolean`	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`	Flag to enable/disable SSL communication with Gateway, default false
`storageMode`	`string`	Indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.

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

.spec.config.volumes[].scaleIO.secretRef

Description

SecretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.config.volumes[].secret

Description

Secret represents a secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	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`	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`	Specify whether the Secret or its keys must be defined
`secretName`	`string`	Name of the secret in the pod's namespace to use. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/storage/volumes#secret ↗

.spec.config.volumes[].secret.items

Description

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.config.volumes[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	The key to project.

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`	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.config.volumes[].storageos

Description

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

Type

`object`

Property	Type	Description
`fsType`	`string`	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
`readOnly`	`boolean`	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.config.volumes[].storageos.secretRef

Description

SecretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?

.spec.config.volumes[].vsphereVolume

Description

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

Type

`object`

Required

`volumePath`

Property	Type	Description
`fsType`	`string`	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.
`storagePolicyID`	`string`	Storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.

Property	Type	Description
`storagePolicyName`	`string`	Storage Policy Based Management (SPBM) profile name.
`volumePath`	`string`	Path that identifies vSphere volume vmdk

.status

Type

`object`

Required

`lastUpdated`

Property	Type	Description
`catalogHealth`	`array`	CatalogHealth contains the Subscription's view of its relevant CatalogSources' status. It is used to determine SubscriptionStatusConditions related to CatalogSources.
`conditions`	`array`	Conditions is a list of the latest available observations about a Subscription's current state.
`currentCSV`	`string`	CurrentCSV is the CSV the Subscription is progressing to.

Property	Type	Description
`installPlanGeneration`	`integer`	InstallPlanGeneration is the current generation of the installplan
`installPlanRef`	`object`	InstallPlanRef is a reference to the latest InstallPlan that contains the Subscription's current CSV.
`installedCSV`	`string`	InstalledCSV is the CSV currently installed by the Subscription.
`installplan`	`object`	Install is a reference to the latest InstallPlan generated for the Subscription. DEPRECATED: InstallPlanRef
`lastUpdated`	`string`	LastUpdated represents the last time that the Subscription status was updated.
`reason`	`string`	Reason is the reason the Subscription was transitioned to its current state.
`state`	`string`	State represents the current state of the Subscription

.status.catalogHealth

Description

CatalogHealth contains the Subscription's view of its relevant CatalogSources' status. It is used to determine SubscriptionStatusConditions related to CatalogSources.

Type

`array`

.status.catalogHealth[]

Description

SubscriptionCatalogHealth describes the health of a CatalogSource the Subscription knows about.

Type

`object`

Required

`catalogSourceRef` `healthy` `lastUpdated`

Property	Type	Description
`catalogSourceRef`	`object`	CatalogSourceRef is a reference to a CatalogSource.
`healthy`	`boolean`	Healthy is true if the CatalogSource is healthy; false otherwise.
`lastUpdated`	`string`	LastUpdated represents the last time that the CatalogSourceHealth changed

.status.catalogHealth[].catalogSourceRef

Description

CatalogSourceRef is a reference to a CatalogSource.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

Property	Type	Description
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.status.conditions

Description

Conditions is a list of the latest available observations about a Subscription's current state.

Type

`array`

.status.conditions[]

Description

SubscriptionCondition represents the latest available observations of a Subscription's state.

Type

`object`

Required

`status` `type`

Property	Type	Description
`lastHeartbeatTime`	`string`	LastHeartbeatTime is the last time we got an update on a given condition
`lastTransitionTime`	`string`	LastTransitionTime is the last time the condition transit from one status to another
`message`	`string`	Message is a human-readable message indicating details about last transition.
`reason`	`string`	Reason is a one-word CamelCase reason for the condition's last transition.
`status`	`string`	Status is the status of the condition, one of True, False, Unknown.
`type`	`string`	Type is the type of Subscription condition.

.status.installPlanRef

Description

InstallPlanRef is a reference to the latest InstallPlan that contains the Subscription's current CSV.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

Property	Type	Description
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.status.installplan

Description

Install is a reference to the latest InstallPlan generated for the Subscription. DEPRECATED: InstallPlanRef

Type

`object`

Required

`apiVersion` `kind` `name` `uuid`

Property	Type	Description
`apiVersion`	`string`	

Property	Type	Description
`kind`	`string`	
`name`	`string`	
`uuid`	`string`	UID is a type that holds unique ID values, including UUIDs. Because we don't ONLY use UUIDs, this is an alias to string. Being a type captures intent and helps make sure that UIDs and names do not get conflated.

API Endpoints

The following API endpoints are available:

- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions`
  - `DELETE`: delete collection of Subscription
  - `GET`: list objects of kind Subscription
  - `POST`: create a new Subscription
- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions/{name}`
  - `DELETE`: delete the specified Subscription
  - `GET`: read the specified Subscription
  - `PATCH`: partially update the specified Subscription
  - `PUT`: replace the specified Subscription
- `/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions/{name}/status`
  - `GET`: read status of the specified Subscription
  - `PATCH`: partially update status of the specified Subscription

- `PUT`: replace status of the specified Subscription

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions

HTTP method

`DELETE`

Description

delete collection of Subscription

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind Subscription

HTTP responses

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

HTTP method

`POST`

Description

create a new Subscription

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`	`Subscription` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Subscription` schema

HTTP code	Response body
201 - Created	`Subscription` schema
202 - Accepted	`Subscription` schema
401 - Unauthorized	Empty

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions/{name}

HTTP method

`DELETE`

Description

delete the specified Subscription

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 Subscription

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified Subscription

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

Parameter	Type	Description
		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	`Subscription` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Subscription

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

Parameter	Type	Description
		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`	`Subscription` schema	`application/json` formatted

HTTP responses

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

/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions/{name}/status

HTTP method

`GET`

Description

read status of the specified Subscription

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified Subscription

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	`Subscription` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Subscription

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`	`Subscription` schema	`application/json` formatted

HTTP responses

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

Operator [operators.operators.coreos.com/v1]

Description

Operator represents a cluster operator.

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 ↗
`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:

Property	Type	Description
		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`	OperatorSpec defines the desired state of Operator
`status`	`object`	OperatorStatus defines the observed state of an Operator and its components

.spec

Description

OperatorSpec defines the desired state of Operator

Type

`object`

.status

Description

OperatorStatus defines the observed state of an Operator and its components

Type

`object`

Property	Type	Description
`components`	`object`	Components describes resources that compose the operator.

.status.components

Description

Components describes resources that compose the operator.

Type

`object`

Required

`labelSelector`

Property	Type	Description
`labelSelector`	`object`	LabelSelector is a label query over a set of resources used to select the operator's components
`refs`	`array`	Refs are a set of references to the operator's component resources, selected with LabelSelector.

.status.components.labelSelector

Description

LabelSelector is a label query over a set of resources used to select the operator's components

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.

.status.components.labelSelector.matchExpressions

Description

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

Type

`array`

.status.components.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.

.status.components.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`

.status.components.labelSelector.matchExpressions[].values[]

Type

`string`

.status.components.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`

.status.components.refs

Description

Refs are a set of references to the operator's component resources, selected with LabelSelector.

Type

`array`

.status.components.refs[]

Description

RichReference is a reference to a resource, enriched with its status conditions.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`conditions`	`array`	Conditions represents the latest state of the component.

Property	Type	Description
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids↗

.status.components.refs[].conditions

Description

Conditions represents the latest state of the component.

Type

`array`

.status.components.refs[].conditions[]

Description

Condition represent the latest available observations of an component's state.

Type

`object`

Required

`status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	Last time the condition transitioned from one status to another.

Property	Type	Description
`lastUpdateTime`	`string`	Last time the condition was probed
`message`	`string`	A human readable message indicating details about the transition.
`reason`	`string`	The reason for the condition's last transition.
`status`	`string`	Status of the condition, one of True, False, Unknown.
`type`	`string`	Type of condition.

API Endpoints

The following API endpoints are available:

- `/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators`
  - `DELETE` : delete collection of Operator
  - `GET` : list objects of kind Operator
  - `POST` : create a new Operator
- `/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators/{name}`
  - `DELETE` : delete the specified Operator
  - `GET` : read the specified Operator

- `PATCH`: partially update the specified Operator
- `PUT`: replace the specified Operator

- `/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators/{name}/status`
  - `GET`: read status of the specified Operator
  - `PATCH`: partially update status of the specified Operator
  - `PUT`: replace status of the specified Operator

/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators

HTTP method

`DELETE`

Description

delete collection of Operator

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind Operator

HTTP responses

HTTP code	Response body
200 - OK	`OperatorList` schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new Operator

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`	`Operator` schema	`application/json` formatted

HTTP responses

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

/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators/{name}

HTTP method

`DELETE`

Description

delete the specified Operator

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 Operator

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified Operator

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:

Parameter	Type	Description
		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	`Operator` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Operator

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`	`Operator` schema	`application/json` formatted

HTTP responses

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

/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators/{name}/status

HTTP method

`GET`

Description

read status of the specified Operator

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified Operator

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	`Operator` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Operator

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`	`Operator` schema	`application/json` formatted

HTTP responses

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