OpenShift Container Platform 4.18

OperatorHub APIs

Reference guide for OperatorHub APIs

OpenShift Container Platform 4.18 OperatorHub APIs

Reference guide for OperatorHub APIs

Legal Notice

Abstract

This document describes the OpenShift Container Platform OperatorHub API objects and their detailed specifications.

Table of Contents

CHAPTER 1. OPERATORHUB APIS

1.1. CATALOGSOURCE [OPERATORS.COREOS.COM/V1ALPHA1]

Description

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

Type

object

1.2. CLUSTERSERVICEVERSION [OPERATORS.COREOS.COM/V1ALPHA1]

Description

ClusterServiceVersion is a Custom Resource of type **ClusterServiceVersionSpec**.

Type

object

1.3. INSTALLPLAN [OPERATORS.COREOS.COM/V1ALPHA1]

Description

InstallPlan defines the installation of a set of operators.

Type

object

1.4. OLMCONFIG [OPERATORS.COREOS.COM/V1]

Description

OLMConfig is a resource responsible for configuring OLM.

Type

object

1.5. OPERATOR [OPERATORS.COREOS.COM/V1]

Description

Operator represents a cluster operator.

Type

object

1.6. OPERATORCONDITION [OPERATORS.COREOS.COM/V2]

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

1.7. 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

1.8. PACKAGEMANIFEST [PACKAGES.OPERATORS.COREOS.COM/V1]

Description

PackageManifest holds information about a package, which is a reference to one (or more) channels under a single package.

Type

object

1.9. SUBSCRIPTION [OPERATORS.COREOS.COM/V1ALPHA1]

Description

Subscription keeps operators up to date by tracking changes to Catalogs.

Type

object

CHAPTER 2. CATALOGSOURCE [OPERATORS.COREOS.COM/V1ALPHA1]

Description

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

Type

object

Required

- **metadata**

- **spec**

2.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	

Property	Type	Description
status	**object**	

2.1.1. .spec

Description

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: <registry-host or ip>:<port> Only used when SourceType = SourceTypeGrpc. Ignored when the Image field is set.
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**	

Property	Type	Description
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 values will be ranked lexicographically based on its name.
publisher	**string**	
runAsRoot	**boolean**	RunAsRoot allows admins to indicate that they wish to run the CatalogSource pod in a privileged pod as root. This should only be enabled when running older catalog images which could not be run as non-root.
secrets	**array (string)**	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

Property	Type	Description
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

2.1.2. .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
affinity	**object**	Affinity is the catalog source's pod's affinity.
extractContent	**object**	ExtractContent configures the gRPC catalog Pod to extract catalog metadata from the provided index image and use a well-known version of the **opm** server to expose it. The catalog index image that this CatalogSource is configured to use **must** be using the file-based catalogs in order to utilize this feature.

Property	Type	Description
memoryTarget	**integer-or-string**	MemoryTarget configures the $GOMEMLIMIT value for the gRPC catalog Pod. This is a soft memory limit for the server, which the runtime will attempt to meet but makes no guarantees that it will do so. If this value is set, the Pod will have the following modifications made to the container running the server: - the $GOMEMLIMIT environment variable will be set to this value in bytes - the memory request will be set to this value

This field should be set if it's desired to reduce the footprint of a catalog server as much as possible, or if a catalog being served is very large and needs more than the default allocation. If your index image has a file-system cache, determine a good approximation for this value by doubling the size of the package cache at /tmp/cache/cache/packages.json in the index image.

This field is best-effort; if unset, no default will be used and no Pod memory limit or $GOMEMLIMIT value will be set. |
| **nodeSelector** | **object (string)** | 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. |
| **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. |

Property	Type	Description
securityContextConfig	**string**	SecurityContextConfig can be one of **legacy** or **restricted**. The CatalogSource's pod is either injected with the right pod.spec.securityContext and pod.spec.container[*].securityContext values to allow the pod to run in Pod Security Admission (PSA) **restricted** mode, or doesn't set these values at all, in which case the pod can only be run in PSA **baseline** or **privileged** namespaces. If the SecurityContextConfig is unspecified, the mode will be determined by the namespace's PSA configuration. If the namespace is enforcing **restricted** mode, then the pod will be configured as if **restricted** was specified. Otherwise, it will be configured as if **legacy** was specified. Specifying a value other than **legacy** or **restricted** result in a validation error. When using older catalog images, which can not run in **restricted** mode, the SecurityContextConfig should be set to **legacy**. More information about PSA can be found here: https://kubernetes.io/docs/concepts/security/pod-security-admission/'
tolerations	**array**	Tolerations are the catalog source's pod's tolerations.
tolerations[]	**object**	The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

2.1.3. .spec.grpcPodConfig.affinity

Description

    Affinity is the catalog source's pod's affinity.

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)).

2.1.4. .spec.grpcPodConfig.affinity.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	**array**	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, 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.

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution[]	**object**	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).
requiredDuringSchedulingIgnoredDuringExecution	**object**	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.

2.1.5. .spec.grpcPodConfig.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

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

2.1.6. .spec.grpcPodConfig.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

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.

2.1.7. .spec.grpcPodConfig.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

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.
matchExpressions[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	**array**	A list of node selector requirements by node's fields.
matchFields[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

2.1.8. .spec.grpcPodConfig.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

Description

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

Type

array

2.1.9. .spec.grpcPodConfig.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

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 (string)**	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.

2.1.10. .spec.grpcPodConfig.affinity.nodeAffinity.preferredDuringSchedulingIgnoredD

Description

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

Type

array

2.1.11. .spec.grpcPodConfig.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

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 (string)**	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.

2.1.12. .spec.grpcPodConfig.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

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.
nodeSelectorTerms[]	**object**	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.

2.1.13. .spec.grpcPodConfig.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

Description

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

Type

array

2.1.14. .spec.grpcPodConfig.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

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.
matchExpressions[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	**array**	A list of node selector requirements by node's fields.
matchFields[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

2.1.15. .spec.grpcPodConfig.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

Description

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

Type

array

2.1.16. .spec.grpcPodConfig.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

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 (string)**	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.

2.1.17. .spec.grpcPodConfig.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

Description

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

Type

array

2.1.18. .spec.grpcPodConfig.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

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

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 (string)**	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.

2.1.19. .spec.grpcPodConfig.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

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	**array**	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "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.
preferredDuringSchedulingIgnoredDuringExecution[]	**object**	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	**array**	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to 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.

Property	Type	Description
requiredDuringSchedulingIg noredDuringExecution[]	**object**	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

2.1.20. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

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

2.1.21. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

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.

Property	Type	Description
weight	**integer**	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

2.1.22. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

Description

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

Type

object

Required

- **topologyKey**

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

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

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

2.1.23. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredD

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

2.1.24. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredD

Description

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

Type

array

2.1.25. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

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 (string)	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.

2.1.26. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

Description

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

Type

object

Property	Type	Description

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

2.1.27. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

Description

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

Type

array

2.1.28. .spec.grpcPodConfig.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

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 (string)**	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.

2.1.29. .spec.grpcPodConfig.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

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

2.1.30. .spec.grpcPodConfig.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

Description

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

Type

object

Required

- **topologyKey**

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

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

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

2.1.31. .spec.grpcPodConfig.affinity.podAffinity.requiredDuringSchedulingIgnoredDuri

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

2.1.32. .spec.grpcPodConfig.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

Description

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

Type

array

2.1.33. .spec.grpcPodConfig.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

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 (string)**	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.

2.1.34. .spec.grpcPodConfig.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

2.1.35. .spec.grpcPodConfig.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

Description

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

Type

array

2.1.36. .spec.grpcPodConfig.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

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 (string)**	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.

2.1.37. .spec.grpcPodConfig.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

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	**array**	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.
preferredDuringSchedulingIgnoredDuringExecution[]	**object**	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	**array**	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.

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution[]	**object**	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

2.1.38. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

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

2.1.39. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

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.

Property	Type	Description
weight	**integer**	weight associated with matching the corresponding podAffinityTerm, in the range 1–100.

2.1.40. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

Description

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

Type

object

Required

- **topologyKey**

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

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

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

2.1.41. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

2.1.42. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

Description

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

Type

array

2.1.43. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

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 (string)**	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.

2.1.44. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

Description

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

Type

object

Property	Type	Description

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

2.1.45. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

Description

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

Type

array

2.1.46. .spec.grpcPodConfig.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

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 (string)**	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.

2.1.47. .spec.grpcPodConfig.affinity.podAntiAffinity.requiredDuringSchedulingIgnorec

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

2.1.48. .spec.grpcPodConfig.affinity.podAntiAffinity.requiredDuringSchedulingIgnorec

Description

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

Type

object

Required

- **topologyKey**

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

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

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

2.1.49. .spec.grpcPodConfig.affinity.podAntiAffinity.requiredDuringSchedulingIgnored

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

2.1.50. .spec.grpcPodConfig.affinity.podAntiAffinity.requiredDuringSchedulingIgnored

Description

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

Type

array

2.1.51. .spec.grpcPodConfig.affinity.podAntiAffinity.requiredDuringSchedulingIgnored

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 (string)**	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.

2.1.52. .spec.grpcPodConfig.affinity.podAntiAffinity.requiredDuringSchedulingIgnorec

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

2.1.53. .spec.grpcPodConfig.affinity.podAntiAffinity.requiredDuringSchedulingIgnorec

Description

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

Type

array

2.1.54. .spec.grpcPodConfig.affinity.podAntiAffinity.requiredDuringSchedulingIgnored

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 (string)**	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.

2.1.55. .spec.grpcPodConfig.extractContent

Description

ExtractContent configures the gRPC catalog Pod to extract catalog metadata from the provided index image and use a well-known version of the **opm** server to expose it. The catalog index image that this CatalogSource is configured to use **must** be using the file-based catalogs in order to utilize this feature.

Type

object

Required

- **cacheDir**
- **catalogDir**

Property	Type	Description
cacheDir	**string**	CacheDir is the directory storing the pre-calculated API cache.
catalogDir	**string**	CatalogDir is the directory storing the file-based catalog contents.

2.1.56. .spec.grpcPodConfig.tolerations

Description

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

Type

array

2.1.57. .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.

Property	Type	Description
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.

2.1.58. .spec.icon

Description

Type

object

Required

- **base64data**

- **mediatype**

Property	Type	Description
base64data	**string**	
mediatype	**string**	

2.1.59. .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**	

2.1.60. .spec.updateStrategy.registryPoll

Description

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.

2.1.61. .status

Description

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.
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.

Property	Type	Description
configMapReference	**object**	ConfigMapReference (deprecated) is the reference to the ConfigMap containing the catalog source's configuration, when the catalog source is a ConfigMap
connectionState	**object**	ConnectionState represents the current state of the CatalogSource's connection to the registry
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**	RegistryService represents the current state of the GRPC service used to serve the catalog

2.1.62. .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

2.1.63. .status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource.

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.
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.

2.1.64. .status.configMapReference

Description

ConfigMapReference (deprecated) is the reference to the ConfigMap containing the catalog source's configuration, when the catalog source is a ConfigMap

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.

2.1.65. .status.connectionState

Description

ConnectionState represents the current state of the CatalogSource's connection to the registry

Type

object

Required

- **lastObservedState**

Property	Type	Description
address	**string**	
lastConnect	**string**	

Property	Type	Description
lastObservedState	**string**	

2.1.66. .status.registryService

Description

RegistryService represents the current state of the GRPC service used to serve the catalog

Type

object

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

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/operators.coreos.com/v1alpha1/catalogsources**

  - **GET**: list objects of kind CatalogSource

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

  - **DELETE**: delete collection of CatalogSource

  - **GET**: list objects of kind CatalogSource

  - **POST**: create a CatalogSource

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

  - **DELETE**: delete a 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

2.2.1. /apis/operators.coreos.com/v1alpha1/catalogsources

HTTP method

   GET

Description

   list objects of kind CatalogSource

Table 2.1. HTTP responses

HTTP code	Reponse body
200 - OK	**CatalogSourceList** schema
401 – Unauthorized	Empty

2.2.2. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsource

HTTP method

   DELETE

Description

   delete collection of CatalogSource

Table 2.2. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
401 – Unauthorized	Empty

HTTP method

   GET

Description

   list objects of kind CatalogSource

Table 2.3. HTTP responses

HTTP code	Reponse body
200 – OK	**CatalogSourceList** schema
401 – Unauthorized	Empty

HTTP method

> **POST**

Description

> create a CatalogSource

Table 2.4. 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.

Table 2.5. Body parameters

Parameter	Type	Description
body	**CatalogSource** schema	

Table 2.6. HTTP responses

HTTP code	Reponse body
200 – OK	**CatalogSource** schema
201 – Created	**CatalogSource** schema
202 – Accepted	**CatalogSource** schema
401 – Unauthorized	Empty

2.2.3. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsource

Table 2.7. Global path parameters

Parameter	Type	Description
name	**string**	name of the CatalogSource

HTTP method
   DELETE
Description
   delete a CatalogSource

Table 2.8. 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

Table 2.9. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method
   GET
Description

read the specified CatalogSource

Table 2.10. HTTP responses

HTTP code	Reponse body
200 - OK	**CatalogSource** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified CatalogSource

Table 2.11. 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.

Table 2.12. HTTP responses

HTTP code	Reponse body
200 - OK	**CatalogSource** schema

HTTP code	Reponse body
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace the specified CatalogSource

Table 2.13. 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.

Table 2.14. Body parameters

Parameter	Type	Description
body	**CatalogSource** schema	

Table 2.15. HTTP responses

HTTP code	Reponse body
200 – OK	**CatalogSource** schema
201 – Created	**CatalogSource** schema
401 – Unauthorized	Empty

2.2.4. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/catalogsource

Table 2.16. Global path parameters

Parameter	Type	Description
name	**string**	name of the CatalogSource

HTTP method
    GET

Description
    read status of the specified CatalogSource

Table 2.17. HTTP responses

HTTP code	Reponse body
200 – OK	**CatalogSource** schema
401 – Unauthorized	Empty

HTTP method
    PATCH

Description
    partially update status of the specified CatalogSource

Table 2.18. 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.

Table 2.19. HTTP responses

HTTP code	Reponse body
200 - OK	**CatalogSource** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified CatalogSource

Table 2.20. 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.

Table 2.21. Body parameters

Parameter	Type	Description
body	**CatalogSource** schema	

Table 2.22. HTTP responses

HTTP code	Reponse body
200 – OK	**CatalogSource** schema
201 – Created	**CatalogSource** schema
401 – Unauthorized	Empty

CHAPTER 3. CLUSTERSERVICEVERSION [OPERATORS.COREOS.COM/V1ALPHA1]

Description

ClusterServiceVersion is a Custom Resource of type **ClusterServiceVersionSpec**.

Type

object

Required

- **metadata**

- **spec**

3.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

Property	Type	Description
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.

3.1.1. .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 (string)**	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.

Property	Type	Description
description	**string**	Description of the operator. Can include the features, limitations or use-cases of the operator.
displayName	**string**	The name of the operator in display format.
icon	**array**	The icon for this operator.
icon[]	**object**	
install	**object**	NamedInstallStrategy represents the block of an ClusterServiceVersion resource where the install strategy is specified.
installModes	**array**	InstallModes specify supported installation types
installModes[]	**object**	InstallMode associates an InstallModeType with a flag representing if the CSV supports it
keywords	**array (string)**	A list of keywords describing the operator.
labels	**object (string)**	Map of string keys and values that can be used to organize and categorize (scope and select) objects.
links	**array**	A list of links related to the operator.
links[]	**object**	
maintainers	**array**	A list of organizational entities maintaining the operator.
maintainers[]	**object**	
maturity	**string**	
minKubeVersion	**string**	

Property	Type	Description
nativeAPIs	**array**	
nativeAPIs[]	**object**	GroupVersionKind unambiguously identifies a kind. It doesn't anonymously include GroupVersion to avoid automatic coercion. It doesn't use a GroupVersion to avoid custom marshalling
provider	**object**	The publishing entity behind the operator.
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 only used during catalog creation and plays no part in cluster runtime.
relatedImages[]	**object**	
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 (string)**	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**	
webhookdefinitions	**array**	

Property	Type	Description
webhookdefinitions[]	**object**	WebhookDescription provides details to OLM about required webhooks

3.1.2. .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**	
owned[]	**object**	APIServiceDescription provides details to OLM about apis provided via aggregation
required	**array**	
required[]	**object**	APIServiceDescription provides details to OLM about apis provided via aggregation

3.1.3. .spec.apiservicedefinitions.owned

Description

Type

array

3.1.4. .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**	
actionDescriptors[]	**object**	ActionDescriptor describes a declarative action that can be performed on a custom resource instance
containerPort	**integer**	
deploymentName	**string**	
description	**string**	
displayName	**string**	
group	**string**	
kind	**string**	
name	**string**	
resources	**array**	
resources[]	**object**	APIResourceReference is a reference to a Kubernetes resource type that the referrer utilizes.
specDescriptors	**array**	
specDescriptors[]	**object**	SpecDescriptor describes a field in a spec block of a CRD so that OLM can consume it
statusDescriptors	**array**	
statusDescriptors[]	**object**	StatusDescriptor describes a field in a status block of a CRD so that OLM can consume it
version	**string**	

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

Description

Type

array

3.1.6. .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 (string)**	

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

Description

Type

array

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

Description

APIResourceReference is a reference to a Kubernetes resource type that the referrer utilizes.

Type

object

Required

- **kind**

- **name**

- **version**

Property	Type	Description
kind	**string**	Kind of the referenced resource type.
name	**string**	Plural name of the referenced resource type (CustomResourceDefinition.Spec. Names[].Plural). Empty string if the referenced resource type is not a custom resource.
version	**string**	API Version of the referenced resource type.

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

Description

Type

    array

3.1.10. .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**	

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 (string)**	

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

Description

Type

    array

3.1.12. .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 (string)**	

3.1.13. .spec.apiservicedefinitions.required

Description

Type

array

3.1.14. .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**	
actionDescriptors[]	**object**	ActionDescriptor describes a declarative action that can be performed on a custom resource instance
containerPort	**integer**	
deploymentName	**string**	
description	**string**	
displayName	**string**	
group	**string**	
kind	**string**	
name	**string**	
resources	**array**	

Property	Type	Description
resources[]	**object**	APIResourceReference is a reference to a Kubernetes resource type that the referrer utilizes.
specDescriptors	**array**	
specDescriptors[]	**object**	SpecDescriptor describes a field in a spec block of a CRD so that OLM can consume it
statusDescriptors	**array**	
statusDescriptors[]	**object**	StatusDescriptor describes a field in a status block of a CRD so that OLM can consume it
version	**string**	

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

Description

Type

    array

3.1.16. .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**	

Property	Type	Description
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 (string)**	

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

Description

Type

    array

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

Description

    APIResourceReference is a reference to a Kubernetes resource type that the referrer utilizes.

Type

    object

Required

- **kind**

- **name**

- **version**

Property	Type	Description
kind	**string**	Kind of the referenced resource type.
name	**string**	Plural name of the referenced resource type (CustomResourceDefinition.Spec.Names[].Plural). Empty string if the referenced resource type is not a custom resource.
version	**string**	API Version of the referenced resource type.

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

Description

Type

array

3.1.20. .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**	
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 (string)**	

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

Description

Type

array

3.1.22. .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 (string)**	

3.1.23. .spec.cleanup

Description

Cleanup specifies the cleanup behaviour when the CSV gets deleted

Type

object

Required

- **enabled**

Property	Type	Description
enabled	**boolean**	

3.1.24. .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**	
owned[]	**object**	CRDDescription provides details to OLM about the CRDs
required	**array**	
required[]	**object**	CRDDescription provides details to OLM about the CRDs

3.1.25. .spec.customresourcedefinitions.owned

Description

Type

   array

3.1.26. .spec.customresourcedefinitions.owned[]

Description

   CRDDescription provides details to OLM about the CRDs

Type

   object

Required

- **kind**
- **name**
- **version**

Property	Type	Description
actionDescriptors	**array**	
actionDescriptors[]	**object**	ActionDescriptor describes a declarative action that can be performed on a custom resource instance
description	**string**	
displayName	**string**	
kind	**string**	

Property	Type	Description
name	**string**	
resources	**array**	
resources[]	**object**	APIResourceReference is a reference to a Kubernetes resource type that the referrer utilizes.
specDescriptors	**array**	
specDescriptors[]	**object**	SpecDescriptor describes a field in a spec block of a CRD so that OLM can consume it
statusDescriptors	**array**	
statusDescriptors[]	**object**	StatusDescriptor describes a field in a status block of a CRD so that OLM can consume it
version	**string**	

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

Description

Type

   array

3.1.28. .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**	

Property	Type	Description
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 (string)**	

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

Description

Type

    array

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

Description

    APIResourceReference is a reference to a Kubernetes resource type that the referrer utilizes.

Type

    object

Required

- **kind**
- **name**
- **version**

Property	Type	Description
kind	**string**	Kind of the referenced resource type.
name	**string**	Plural name of the referenced resource type (CustomResourceDefinition.Spec. Names[].Plural). Empty string if the referenced resource type is not a custom resource.

Property	Type	Description
version	**string**	API Version of the referenced resource type.

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

Description

Type

   array

3.1.32. .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 (string)**	

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

Description

Type

   array

3.1.34. .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 (string)**	

3.1.35. .spec.customresourcedefinitions.required

Description

Type

array

3.1.36. .spec.customresourcedefinitions.required[]

Description

CRDDescription provides details to OLM about the CRDs

Type

object

Required

- **kind**

- **name**

- **version**

Property	Type	Description
actionDescriptors	**array**	
actionDescriptors[]	**object**	ActionDescriptor describes a declarative action that can be performed on a custom resource instance
description	**string**	
displayName	**string**	
kind	**string**	
name	**string**	
resources	**array**	
resources[]	**object**	APIResourceReference is a reference to a Kubernetes resource type that the referrer utilizes.
specDescriptors	**array**	
specDescriptors[]	**object**	SpecDescriptor describes a field in a spec block of a CRD so that OLM can consume it
statusDescriptors	**array**	
statusDescriptors[]	**object**	StatusDescriptor describes a field in a status block of a CRD so that OLM can consume it
version	**string**	

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

Description

Type

   array

3.1.38. .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 (string)**	

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

Description

Type

array

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

Description

APIResourceReference is a reference to a Kubernetes resource type that the referrer utilizes.

Type

object

Required

- **kind**

- **name**

- **version**

Property	Type	Description
kind	**string**	Kind of the referenced resource type.
name	**string**	Plural name of the referenced resource type (CustomResourceDefinition.Spec.Names[].Plural). Empty string if the referenced resource type is not a custom resource.
version	**string**	API Version of the referenced resource type.

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

Description

Type

    array

3.1.42. .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.

Property	Type	Description
x-descriptors	array (string)	

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

Description

Type

array

3.1.44. .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 (string)	

3.1.45. .spec.icon

Description

The icon for this operator.

Type

array

3.1.46. .spec.icon[]

Description

Type

object

Required

- **base64data**

- **mediatype**

Property	Type	Description
base64data	**string**	
mediatype	**string**	

3.1.47. .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**	

3.1.48. .spec.install.spec

Description

StrategyDetailsDeployment represents the parsed details of a Deployment InstallStrategy.

Type

object

Required

- **deployments**

Property	Type	Description
clusterPermissions	**array**	
clusterPermissions[]	**object**	StrategyDeploymentPermissions describe the rbac rules and service account needed by the install strategy
deployments	**array**	
deployments[]	**object**	StrategyDeploymentSpec contains the name, spec and labels for the deployment ALM should create
permissions	**array**	
permissions[]	**object**	StrategyDeploymentPermissions describe the rbac rules and service account needed by the install strategy

3.1.49. .spec.install.spec.clusterPermissions

Description

Type

array

3.1.50. .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**	

Property	Type	Description
rules[]	**object**	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.
serviceAccountName	**string**	

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

Description

Type

   array

3.1.52. .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 (string)**	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. "" represents the core API group and "*" represents all API groups.

Property	Type	Description
nonResourceURLs	**array (string)**	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 (string)**	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
resources	**array (string)**	Resources is a list of resources this rule applies to. '*' represents all resources.
verbs	**array (string)**	Verbs is a list of Verbs that apply to ALL the ResourceKinds contained in this rule. '*' represents all verbs.

3.1.53. .spec.install.spec.deployments

Description

Type

array

3.1.54. .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 (string)**	Set is a map of label:value. It implements Labels.
name	**string**	
spec	**object**	DeploymentSpec is the specification of the desired behavior of the Deployment.

3.1.55. .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)
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.

Property	Type	Description
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.
strategy	**object**	The deployment strategy to use to replace existing pods with new ones.
template	**object**	Template describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

3.1.56. .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.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Property	Type	Description
matchLabels	**object (string)**	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.

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

Description

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

Type

> **array**

3.1.58. .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 (string)**	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.

3.1.59. .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.
type	**string**	Type of deployment. Can be "Recreate" or "RollingUpdate". Default is RollingUpdate.

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

Description

Rolling update config params. Present only if DeploymentStrategyType = RollingUpdate.

Type

object

Property	Type	Description

Property	Type	Description
maxSurge	**integer-or-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-or-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.

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

Description

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

Type

object

Property	Type	Description
metadata	`` `` ``	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

3.1.62. .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 be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer.
affinity	**object**	If specified, the pod's scheduling constraints
automountServiceAccountToken	**boolean**	AutomountServiceAccountToken indicates whether a service account token should be automatically mounted.

Property	Type	Description
containers	**array**	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.
containers[]	**object**	A single application container that you want to run within a pod.
dnsConfig	**object**	Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy.
dnsPolicy	**string**	Set DNS policy for the pod. Defaults to "ClusterFirst". Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' or 'None'. DNS parameters given in DNSConfig will be merged with the policy selected with DNSPolicy. To have DNS options set along with hostNetwork, you have to specify DNS policy explicitly to 'ClusterFirstWithHostNet'.
enableServiceLinks	**boolean**	EnableServiceLinks indicates whether information about services should be injected into pod's environment variables, matching the syntax of Docker links. Optional: Defaults to true.
ephemeralContainers	**array**	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.

Property	Type	Description
ephemeralContainers[]	**object**	An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.
hostAliases	**array**	HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
hostAliases[]	**object**	HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.
hostIPC	**boolean**	Use the host's ipc namespace. Optional: Default to false.

Property	Type	Description
hostNetwork	**boolean**	Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified. Default to false.
hostPID	**boolean**	Use the host's pid namespace. Optional: Default to false.
hostUsers	**boolean**	Use the host's user namespace. Optional: Default to true. If set to true or not present, the pod will be run in the host user namespace, useful for when the pod needs a feature only available to the host user namespace, such as loading a kernel module with CAP_SYS_MODULE. When set to false, a new userns is created for the pod. Setting false is useful for mitigating container breakout vulnerabilities even allowing users to run their containers as root without actually having root privileges on the host. This field is alpha-level and is only honored by servers that enable the UserNamespacesSupport feature.
hostname	**string**	Specifies the hostname of the Pod If not specified, the pod's hostname will be set to a system-defined value.
imagePullSecrets	**array**	ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info: https://kubernetes.io/docs/conc epts/containers/images#specifyi ng-imagepullsecrets-on-a-pod
imagePullSecrets[]	**object**	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Property	Type	Description
initContainers	**array**	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/concepts/workloads/pods/init-containers/
initContainers[]	**object**	A single application container that you want to run within a pod.
nodeName	**string**	NodeName indicates in which node this pod is scheduled. If empty, this pod is a candidate for scheduling by the scheduler defined in schedulerName. Once this field is set, the kubelet for this node becomes responsible for the lifecycle of this pod. This field should not be used to express a desire for the pod to be scheduled on a specific node. https://kubernetes.io/concepts/scheduling-eviction/assign-pod-node/#nodename

Property	Type	Description
nodeSelector	**object (string)**	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/

Property	Type	Description
os	**object**	Specifies the OS of the containers in the pod. Some pod and container fields are restricted if this is set. If the OS field is set to linux, the following fields must be unset: - securityContext.windowsOptions If the OS field is set to windows, following fields must be unset: - spec.hostPID - spec.hostIPC - spec.hostUsers - spec.securityContext.appArmorProfile - spec.securityContext.seLinuxOptions - spec.securityContext.seccompProfile - spec.securityContext.fsGroup - spec.securityContext.fsGroupChangePolicy - spec.securityContext.sysctls - spec.shareProcessNamespace - spec.securityContext.runAsUser - spec.securityContext.runAsGroup - spec.securityContext.supplementalGroups - spec.securityContext.supplementalGroupsPolicy - spec.containers[**]**.securityContext.appArmorProfile** - spec.containers[].securityContext.seLinuxOptions - spec.containers[**]**.securityContext.seccompProfile** - spec.containers[].securityContext.capabilities - spec.containers[**]**.securityContext.readOnlyRootFilesystem** - spec.containers[].securityContext.privileged - spec.containers[**]**.securityContext.allowPrivilegeEscalation** - spec.containers[].securityContext.procMount - spec.containers[**]**.securityContext.runAsUser** - spec.containers[].securityContext.runAsGroup
os	**object**	

Property	Type	Description
overhead	**integer-or-string**	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
preemptionPolicy	**string**	PreemptionPolicy is the Policy for preempting pods with lower priority. One of Never, PreemptLowerPriority. Defaults to PreemptLowerPriority if unset.
priority	**integer**	The priority value. Various system components use this field to find the priority of the pod. When Priority Admission Controller is enabled, it prevents users from setting this field. The admission controller populates this field from PriorityClassName. The higher the value, the higher the priority.

Property	Type	Description
priorityClassName	**string**	If specified, indicates the pod's priority. "system-node-critical" and "system-cluster-critical" are two special keywords which indicate the highest priorities with the former being the highest priority. Any other name must be defined by creating a PriorityClass object with that name. If not specified, the pod priority will be default or zero if there is no default.
readinessGates	**array**	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
readinessGates[]	**object**	PodReadinessGate contains the reference to a pod condition
resourceClaims	**array**	ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.

Property	Type	Description
resourceClaims[]	**object**	PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod. It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.
restartPolicy	**string**	Restart policy for all containers within the pod. One of Always, OnFailure, Never. In some contexts, only a subset of those values may be permitted. Default to Always. More info: https://kubernetes.io/concepts/workloads/pods/pod-lifecycle/#restart-policy
runtimeClassName	**string**	RuntimeClassName refers to a RuntimeClass object in the node.k8s.io group, which should be used to run this pod. If no RuntimeClass resource matches the named class, the pod will not be run. If unset or empty, the "legacy" RuntimeClass will be used, which is an implicit class with an empty definition that uses the default runtime handler. More info: https://git.k8s.io/enhancements/keps/sig-node/585-runtime-class
schedulerName	**string**	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.

Property	Type	Description
schedulingGates	**array**	SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.
schedulingGates[]	**object**	PodSchedulingGate is associated to a Pod to guard its scheduling.
securityContext	**object**	SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See type description for default values of each field.
serviceAccount	**string**	DeprecatedServiceAccount is a deprecated alias for ServiceAccountName. Deprecated: Use serviceAccountName instead.
serviceAccountName	**string**	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks /configure-pod-container/configure-service-account/

Property	Type	Description
setHostnameAsFQDN	**boolean**	If true the pod's hostname will be configured as the pod's FQDN, rather than the leaf name (the default). In Linux containers, this means setting the FQDN in the hostname field of the kernel (the nodename field of struct utsname). In Windows containers, this means setting the registry value of hostname for the registry key HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\Tcpip\\Parameters to FQDN. If a pod does not have FQDN, this has no effect. Default to false.
shareProcessNamespace	**boolean**	Share a single process namespace between all of the containers in a pod. When this is set containers will be able to view and signal processes from other containers in the same pod, and the first process in each container will not be assigned PID 1. HostPID and ShareProcessNamespace cannot both be set. Optional: Default to false.
subdomain	**string**	If specified, the fully qualified Pod hostname will be "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>". If not specified, the pod will not have a domainname at all.

Property	Type	Description
terminationGracePeriodSeconds	**integer**	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. Defaults to 30 seconds.
tolerations	**array**	If specified, the pod's tolerations.
tolerations[]	**object**	The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.
topologySpreadConstraints	**array**	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.
topologySpreadConstraints[]	**object**	TopologySpreadConstraint specifies how to spread matching pods among the given topology.
volumes	**array**	List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	**object**	Volume represents a named volume in a pod that may be accessed by any container in the pod.

3.1.63. .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)).

3.1.64. .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 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.

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution[]	**object**	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).
requiredDuringSchedulingIgnoredDuringExecution	**object**	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.

3.1.65. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.prefe

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

3.1.66. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.prefe

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.

3.1.67. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.prefe

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.
matchExpressions[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	**array**	A list of node selector requirements by node's fields.
matchFields[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

3.1.68. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.prefe

Description

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

Type

array

3.1.69. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.prefe

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 (string)**	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.

3.1.70. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.prefe

Description

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

Type

array

3.1.71. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.prefer

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 (string)**	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.

3.1.72. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requi

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.
nodeSelectorTerms[]	**object**	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.

3.1.73. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requir

Description

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

Type

array

3.1.74. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requi

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.
matchExpressions[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	**array**	A list of node selector requirements by node's fields.
matchFields[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

3.1.75. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requi

Description

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

Type

array

3.1.76. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requi

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 (string)**	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.

3.1.77. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requi

Description

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

Type

array

3.1.78. .spec.install.spec.deployments[].spec.template.spec.affinity.nodeAffinity.requi

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 (string)**	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.

3.1.79. .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

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	**array**	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "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.
preferredDuringSchedulingIgnoredDuringExecution[]	**object**	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	**array**	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to 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.

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution[]	**object**	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

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

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

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

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.

Property	Type	Description
weight	**integer**	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

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

Description

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

Type

object

Required

- **topologyKey**

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

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

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

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

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

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

Description

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

Type

array

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

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 (string)**	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.

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

Description

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

Type

object

Property	Type	Description

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

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

Description

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

Type

array

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

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 (string)**	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.

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

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

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

Description

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

Type

object

Required

- **topologyKey**

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

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

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

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

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

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

Description

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

Type

array

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

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 (string)**	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.

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

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

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

Description

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

Type

array

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

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 (string)**	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.

3.1.97. .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

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	**array**	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.
preferredDuringSchedulingIgnoredDuringExecution[]	**object**	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	**array**	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.

Property	Type	Description
requiredDuringSchedulingIg noredDuringExecution[]	**object**	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

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

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**

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

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.

Property	Type	Description
weight	**integer**	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

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

Description

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

Type

object

Required

- **topologyKey**

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

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

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

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

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

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

Description

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

Type

array

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

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 (string)**	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.

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

Description

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

Type

object

Property	Type	Description

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

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

Description

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

Type

array

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

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 (string)**	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.

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

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**

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

Description

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

Type

> **object**

Required

> - **topologyKey**

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

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

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

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

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

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

Description

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

Type

array

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

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 (string)**	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.

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

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

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

Description

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

Type

array

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

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 (string)**	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.

3.1.115. .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

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

Property	Type	Description
env[]	**object**	EnvVar represents an environment variable present in a Container.
envFrom	**array**	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.
envFrom[]	**object**	EnvFromSource represents the source of a set of ConfigMaps
image	**string**	Container image name. More info: [https://kubernetes.io/docs/conc epts/containers/images](https://kubernetes.io/docs/concepts/containers/images) This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.
imagePullPolicy	**string**	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: [https://kubernetes.io/docs/conc epts/containers/images#updatin g-images](https://kubernetes.io/docs/concepts/containers/images#updating-images)
lifecycle	**object**	Actions that the management system should take in response to container lifecycle events. Cannot be updated.

Property	Type	Description
livenessProbe	**object**	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
name	**string**	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
ports	**array**	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255. Cannot be updated.
ports[]	**object**	ContainerPort represents a network port in a single container.
readinessProbe	**object**	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
resizePolicy	**array**	Resources resize policy for the container.
resizePolicy[]	**object**	ContainerResizePolicy represents resource resize policy for the container.

Property	Type	Description
resources	**object**	Compute Resources required by this container. Cannot be updated. More info: [https://kubernetes.io/docs/conc epts/configuration/manage-resources-containers/](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
restartPolicy	**string**	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
securityContext	**object**	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/](https://kubernetes.io/docs/tasks/configure-pod-container/security-context/)

Property	Type	Description
startupProbe	**object**	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
stdin	**boolean**	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	**boolean**	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false

Property	Type	Description
terminationMessagePath	**string**	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.
terminationMessagePolicy	**string**	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated.
tty	**boolean**	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	**array**	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	**object**	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	**array**	Pod volumes to mount into the container's filesystem. Cannot be updated.

Property	Type	Description
volumeMounts[]	**object**	VolumeMount describes a mounting of a Volume within a container.
workingDir	**string**	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

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

Description

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

Type

array

3.1.118. .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.

Property	Type	Description
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.

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

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

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

Description

Selects a key of a ConfigMap.

Type

object

Required

- **key**

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

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

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.

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

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-or-string**	Specifies the output format of the exposed resources, defaults to "1"
resource	**string**	Required: resource to select

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

Description

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

Type

object

Required

- **key**

- key

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

3.1.124. .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

3.1.125. .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

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

Description

The ConfigMap to select from

Type

object

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

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

Description

The Secret to select from

Type

object

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

3.1.128. .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 Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

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

Description

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

Type

object

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	**array (string)**	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.

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

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

Type

object

Required

- **seconds**

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

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

Description

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

Type

object

Required

- **port**

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

Property	Type	Description
port	**integer-or-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.

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

Description

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

Type

object

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	**array (string)**	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.

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

Property	Type	Description
scheme	**string**	Scheme to use for connecting to the host. Defaults to HTTP.

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

Description

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

Type

    array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

    object

Required

- **name**
- **value**

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

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

Description

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

Type

    object

Required

- **seconds**

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

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

Description

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

Type

object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

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

Type

object

Property	Type	Description
exec	**object**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

Property	Type	Description
grpc	**object**	GRPC specifies an action involving a GRPC port.
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	**integer**	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	**object**	TCPSocket specifies an action involving a TCP port.

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description

Property	Type	Description
command	**array (string)**	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.

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

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type

> **object**

Required

> - **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

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

Type

> **array**

3.1.151. .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".

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

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**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

Property	Type	Description
grpc	**object**	GRPC specifies an action involving a GRPC port.
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	**integer**	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	**object**	TCPSocket specifies an action involving a TCP port.

Property	Type	Description
terminationGracePeriodSeco nds	**integer**	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSec onds is used if unset.
timeoutSeconds	**integer**	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/conc epts/workloads/pods/pod-lifecycle#container-probes

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	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.

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

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type
object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

Resources resize policy for the container.

Type
array

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

Description

ContainerResizePolicy represents resource resize policy for the container.

Type
object

Required

- **resourceName**

- **restartPolicy**

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

3.1.161. .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
claims	**array**	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	**object**	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	**integer-or-string**	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

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

3.1.162. .spec.install.spec.deployments[].spec.template.spec.containers[].resources.cla

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

3.1.163. .spec.install.spec.deployments[].spec.template.spec.containers[].resources.cla

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	**string**	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

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

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 Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	**object**	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.
capabilities	**object**	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
privileged	**boolean**	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	**string**	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	**boolean**	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	**integer**	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	**boolean**	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	**object**	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	**object**	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	**object**	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

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

Description

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

Type

object

Required

- **type**

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

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

Description

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

Type

object

Property	Type	Description
add	**array (string)**	Added capabilities
drop	**array (string)**	Removed capabilities

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

Description

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

Type

object

Property	Type	Description
level	**string**	Level is SELinux level label that applies to the container.
role	**string**	Role is a SELinux role label that applies to the container.
type	**string**	Type is a SELinux type label that applies to the container.
user	**string**	User is a SELinux user label that applies to the container.

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

Description

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

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	**string**	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	**string**	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost – a profile defined in a file on the node should be used. RuntimeDefault – the container runtime default profile should be used. Unconfined – no profile should be applied.

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

Description

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

Type

object

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

3.1.170. .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**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	**object**	GRPC specifies an action involving a GRPC port.
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	**integer**	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	**object**	TCPSocket specifies an action involving a TCP port.

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description

Property	Type	Description
command	**array (string)**	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.

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

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

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

Type

array

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

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

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

Description

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

Type

array

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

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **mountPath**

- **name**

Property	Type	Description
mountPath	**string**	Path within the container at which the volume should be mounted. Must not contain ':'.
mountPropagation	**string**	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).

Property	Type	Description
name	**string**	This must match the Name of a Volume.
readOnly	**boolean**	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
recursiveReadOnly	**string**	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	**string**	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

Property	Type	Description
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.

3.1.181. .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 (string)**	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from 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.
options[]	**object**	PodDNSConfigOption defines DNS resolver options of a pod.
searches	**array (string)**	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.

3.1.182. .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

3.1.183. .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**	

3.1.184. .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.

Type

array

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

Description

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

Type

object

Required

- **name**

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

Property	Type	Description
env[]	**object**	EnvVar represents an environment variable present in a Container.
envFrom	**array**	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.
envFrom[]	**object**	EnvFromSource represents the source of a set of ConfigMaps
image	**string**	Container image name. More info: [https://kubernetes.io/docs/conc epts/containers/images](https://kubernetes.io/docs/concepts/containers/images)
imagePullPolicy	**string**	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: [https://kubernetes.io/docs/conc epts/containers/images#updatin g-images](https://kubernetes.io/docs/concepts/containers/images#updating-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 DNS_LABEL. This name must be unique among all containers, init containers and ephemeral containers.
ports	**array**	Ports are not allowed for ephemeral containers.

Property	Type	Description
ports[]	**object**	ContainerPort represents a network port in a single container.
readinessProbe	**object**	Probes are not allowed for ephemeral containers.
resizePolicy	**array**	Resources resize policy for the container.
resizePolicy[]	**object**	ContainerResizePolicy represents resource resize policy for the container.
resources	**object**	Resources are not allowed for ephemeral containers. Ephemeral containers use spare resources already allocated to the pod.
restartPolicy	**string**	Restart policy for the container to manage the restart behavior of each container within a pod. This may only be set for init containers. You cannot set this field on ephemeral containers.
securityContext	**object**	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.
startupProbe	**object**	Probes are not allowed for ephemeral containers.
stdin	**boolean**	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.

Property	Type	Description
stdinOnce	**boolean**	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false
targetContainerName	**string**	If set, the name of the container from PodSpec that this ephemeral container targets. The ephemeral container will be run in the namespaces (IPC, PID, etc) of this container. If not set then the ephemeral container uses the namespaces configured in the Pod spec. The container runtime must implement support for this feature. If the runtime does not support namespace targeting then the result of setting this field is undefined.
terminationMessagePath	**string**	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	**string**	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated.
tty	**boolean**	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	**array**	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	**object**	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	**array**	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.
volumeMounts[]	**object**	VolumeMount describes a mounting of a Volume within a container.
workingDir	**string**	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

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

Description

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

Type

array

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

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.

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

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

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

Description

Selects a key of a ConfigMap.

Type

object

Required

- **key**

Property	Type	Description
key	**string**	The key to select.

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

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

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.

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

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-or-string	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

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

Description

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

Type

object

Required

- **key**

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

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

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

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

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

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

Description

The ConfigMap to select from

Type

object

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

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

Description

The Secret to select from

Type

object

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

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

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/conc epts/containers/container- lifecycle-hooks/#container- hooks](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 Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

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

Description

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

Type

object

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

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

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

Description

Exec specifies the action to take.

Type

> **object**

Property	Type	Description
command	**array (string)**	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.

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

Description

HTTPGet specifies the http request to perform.

Type

> **object**

Required

- **port**

Property	Type	Description

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.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- name

- value

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

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

Description

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

Type

> **object**

Required

- **seconds**

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

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

Description

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

Type

> **object**

Required

- **port**

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

Property	Type	Description
port	**integer-or-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.

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

Description

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

Type

   object

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

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

Description

   Exec specifies the action to take.

Type

   object

Property	Type	Description
command	**array (string)**	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.

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

Property	Type	Description
scheme	**string**	Scheme to use for connecting to the host. Defaults to HTTP.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

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

Type

object

Required

- **seconds**

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

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

Description

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

Type

object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	**object**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	**object**	GRPC specifies an action involving a GRPC port.

Property	Type	Description
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	**integer**	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	**object**	TCPSocket specifies an action involving a TCP port.
terminationGracePeriodSeconds	**integer**	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.

Property	Type	Description
timeoutSeconds	**integer**	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	**array (string)**	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.

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

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.

Property	Type	Description
port	**integer-or-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.

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

Description

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

Type

    array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

    object

Required

- **name**

- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type

    object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

Ports are not allowed for ephemeral containers.

Type

array

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

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.

Property	Type	Description
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".

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

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	**object**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	**object**	GRPC specifies an action involving a GRPC port.
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	**array (string)**	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.

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

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**

- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

Resources resize policy for the container.

Type

array

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

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

- **resourceName**

- **restartPolicy**

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

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

Description

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

Type

object

Property	Type	Description
claims	**array**	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	**object**	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	**integer-or-string**	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

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

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

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

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

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	**string**	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

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

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 Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	**object**	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.
capabilities	**object**	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
privileged	**boolean**	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	**string**	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	**boolean**	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	**integer**	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	**boolean**	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	**object**	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	**object**	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	**object**	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

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

Description

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

Type

object

Required

- **type**

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

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

Description

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

Type

object

Property	Type	Description
add	**array (string)**	Added capabilities
drop	**array (string)**	Removed capabilities

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

Description

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

Type

object

Property	Type	Description
level	**string**	Level is SELinux level label that applies to the container.
role	**string**	Role is a SELinux role label that applies to the container.
type	**string**	Type is a SELinux type label that applies to the container.
user	**string**	User is a SELinux user label that applies to the container.

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

Description

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

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	**string**	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	**string**	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost – a profile defined in a file on the node should be used. RuntimeDefault – the container runtime default profile should be used. Unconfined – no profile should be applied.

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

Description

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

Type

object

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

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

Description

Probes are not allowed for ephemeral containers.

Type

object

Property	Type	Description
exec	**object**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	**object**	GRPC specifies an action involving a GRPC port.
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/conc epts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	**integer**	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	**object**	TCPSocket specifies an action involving a TCP port.

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description

Property	Type	Description
command	**array (string)**	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.

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

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

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

Type

array

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

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

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

Description

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

Type

array

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

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 ':'.

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

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.

3.1.250. .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.

Type

array

3.1.251. .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

Required

- **ip**

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

3.1.252. .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. More info: https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod

Type

array

3.1.253. .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. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.254. .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

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

Property	Type	Description
env	**array**	List of environment variables to set in the container. Cannot be updated.
env[]	**object**	EnvVar represents an environment variable present in a Container.
envFrom	**array**	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.
envFrom[]	**object**	EnvFromSource represents the source of a set of ConfigMaps
image	**string**	Container image name. More info: [https://kubernetes.io/docs/conc epts/containers/images](https://kubernetes.io/docs/concepts/containers/images) This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.
imagePullPolicy	**string**	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: [https://kubernetes.io/docs/conc epts/containers/images#updatin g-images](https://kubernetes.io/docs/concepts/containers/images#updating-images)
lifecycle	**object**	Actions that the management system should take in response to container lifecycle events. Cannot be updated.

Property	Type	Description
livenessProbe	**object**	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
name	**string**	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
ports	**array**	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255. Cannot be updated.
ports[]	**object**	ContainerPort represents a network port in a single container.
readinessProbe	**object**	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
resizePolicy	**array**	Resources resize policy for the container.
resizePolicy[]	**object**	ContainerResizePolicy represents resource resize policy for the container.

Property	Type	Description
resources	**object**	Compute Resources required by this container. Cannot be updated. More info: [https://kubernetes.io/docs/conc epts/configuration/manage-resources-containers/](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
restartPolicy	**string**	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
securityContext	**object**	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/](https://kubernetes.io/docs/tasks/configure-pod-container/security-context/)

Property	Type	Description
startupProbe	**object**	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
stdin	**boolean**	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	**boolean**	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false

Property	Type	Description
terminationMessagePath	**string**	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.
terminationMessagePolicy	**string**	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated.
tty	**boolean**	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	**array**	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	**object**	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	**array**	Pod volumes to mount into the container's filesystem. Cannot be updated.

Property	Type	Description
volumeMounts[]	**object**	VolumeMount describes a mounting of a Volume within a container.
workingDir	**string**	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

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

Description

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

Type

array

3.1.257. .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.

Property	Type	Description
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.

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

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

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

Description

Selects a key of a ConfigMap.

Type

object

Required

- **key**

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

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

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.

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

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-or-string**	Specifies the output format of the exposed resources, defaults to "1"
resource	**string**	Required: resource to select

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

Description

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

Type

object

Required

- **key**

- key

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

3.1.263. .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

3.1.264. .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

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

Description

The ConfigMap to select from

Type

object

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

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

Description

The Secret to select from

Type

object

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

3.1.267. .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 Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

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

Description

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

Type

object

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	**array (string)**	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.

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

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

Type

object

Required

- **seconds**

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

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

Description

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

Type

object

Required

- **port**

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

Property	Type	Description
port	**integer-or-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.

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

Description

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

Type

object

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description
command	**array (string)**	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.

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

Property	Type	Description
scheme	**string**	Scheme to use for connecting to the host. Defaults to HTTP.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**

- **value**

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

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

Description

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

Type

object

Required

- **seconds**

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

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

Description

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

Type

object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

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

Type

object

Property	Type	Description
exec	**object**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

Property	Type	Description
grpc	**object**	GRPC specifies an action involving a GRPC port.
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: [https://kubernetes.io/docs/conc epts/workloads/pods/pod- lifecycle#container-probes](https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes)
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	**integer**	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	**object**	TCPSocket specifies an action involving a TCP port.

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description

Property	Type	Description
command	**array (string)**	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.

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

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**

- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

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

Type

array

3.1.290. .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".

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

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**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

Property	Type	Description
grpc	**object**	GRPC specifies an action involving a GRPC port.
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	**integer**	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	**object**	TCPSocket specifies an action involving a TCP port.

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description

Property	Type	Description
command	**array (string)**	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.

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

Description

> GRPC specifies an action involving a GRPC port.

Type

> **object**

Required

- **port**

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

3.1.295. .spec.install.spec.deployments[].spec.template.spec.initContainers[].readines:

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type

   object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

Resources resize policy for the container.

Type

   array

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

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

   object

Required

- **resourceName**

- **restartPolicy**

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

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

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
claims	**array**	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	**object**	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	**integer-or-string**	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Property	Type	Description
requests	**integer-or-string**	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: [https://kubernetes.io/docs/conc epts/configuration/manage-resources-containers/](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

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

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

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

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	**string**	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

3.1.303. .spec.install.spec.deployments[].spec.template.spec.initContainers[].security(

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 Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	**object**	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.
capabilities	**object**	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
privileged	**boolean**	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	**string**	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	**boolean**	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	**integer**	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	**boolean**	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	**object**	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	**object**	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	**object**	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

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

Description

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

Type

object

Required

- **type**

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

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

Description

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

Type

object

Property	Type	Description
add	**array (string)**	Added capabilities
drop	**array (string)**	Removed capabilities

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

Description

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

Type

object

Property	Type	Description
level	**string**	Level is SELinux level label that applies to the container.
role	**string**	Role is a SELinux role label that applies to the container.
type	**string**	Type is a SELinux type label that applies to the container.
user	**string**	User is a SELinux user label that applies to the container.

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

Description

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

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	**string**	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	**string**	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost – a profile defined in a file on the node should be used. RuntimeDefault – the container runtime default profile should be used. Unconfined – no profile should be applied.

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

Description

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

Type

object

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

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

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**	Exec specifies the action to take.
failureThreshold	**integer**	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	**object**	GRPC specifies an action involving a GRPC port.
httpGet	**object**	HTTPGet specifies the http request to perform.
initialDelaySeconds	**integer**	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	**integer**	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	**integer**	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	**object**	TCPSocket specifies an action involving a TCP port.

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

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

Description

Exec specifies the action to take.

Type

object

Property	Type	Description

Property	Type	Description
command	**array (string)**	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.

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

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

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

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

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.
httpHeaders[]	**object**	HTTPHeader describes a custom header to be used in HTTP probes
path	**string**	Path to access on the HTTP server.
port	**integer-or-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.

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

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

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

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

- **port**

Property	Type	Description
host	**string**	Optional: Host name to connect to, defaults to the pod IP.
port	**integer-or-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.

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

Description

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

Type

array

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

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

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

Description

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

Type

array

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

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **mountPath**

- **name**

Property	Type	Description
mountPath	**string**	Path within the container at which the volume should be mounted. Must not contain ':'.
mountPropagation	**string**	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).

Property	Type	Description
name	**string**	This must match the Name of a Volume.
readOnly	**boolean**	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
recursiveReadOnly	**string**	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	**string**	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

Property	Type	Description
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.

3.1.320. .spec.install.spec.deployments[].spec.template.spec.os

Description

Specifies the OS of the containers in the pod. Some pod and container fields are restricted if this is set.
If the OS field is set to linux, the following fields must be unset: –securityContext.windowsOptions

If the OS field is set to windows, following fields must be unset: – spec.hostPID – spec.hostIPC – spec.hostUsers – spec.securityContext.appArmorProfile – spec.securityContext.seLinuxOptions – spec.securityContext.seccompProfile – spec.securityContext.fsGroup – spec.securityContext.fsGroupChangePolicy – spec.securityContext.sysctls – spec.shareProcessNamespace – spec.securityContext.runAsUser – spec.securityContext.runAsGroup – spec.securityContext.supplementalGroups – spec.securityContext.supplementalGroupsPolicy – spec.containers[**].securityContext.appArmorProfile – spec.containers[**].securityContext.seLinuxOptions – spec.containers[**].securityContext.seccompProfile – spec.containers[**].securityContext.capabilities – spec.containers[**].securityContext.readOnlyRootFilesystem – spec.containers[**].securityContext.privileged – spec.containers[**].securityContext.allowPrivilegeEscalation – spec.containers[**].securityContext.procMount – spec.containers[**].securityContext.runAsUser – spec.containers[**].securityContext.runAsGroup

Type

   object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	**string**	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration Clients should expect to handle additional values and treat unrecognized values in this field as os: null

3.1.321. .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

3.1.322. .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.

3.1.323. .spec.install.spec.deployments[].spec.template.spec.resourceClaims

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name.
This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable.

Type

array

3.1.324. .spec.install.spec.deployments[].spec.template.spec.resourceClaims[]

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod.
It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	Name uniquely identifies this resource claim inside the pod. This must be a DNS_LABEL.
resourceClaimName	**string**	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

Property	Type	Description
resourceClaimTemplateName	**string**	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in pod.status.resourceClaimStatuses. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

3.1.325. .spec.install.spec.deployments[].spec.template.spec.schedulingGates

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod.
SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

array

3.1.326. .spec.install.spec.deployments[].spec.template.spec.schedulingGates[]

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	Name of the scheduling gate. Each scheduling gate must have a unique name field.

3.1.327. .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
appArmorProfile	**object**	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	**integer**	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
fsGroupChangePolicy	**string**	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	**integer**	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	**boolean**	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Property	Type	Description
runAsUser	**integer**	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	**object**	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	**object**	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
supplementalGroups	**array (integer)**	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
supplementalGroupsPolicy	**string**	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
sysctls	**array**	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
sysctls[]	**object**	Sysctl defines a kernel parameter to be set
windowsOptions	**object**	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

3.1.328. .spec.install.spec.deployments[].spec.template.spec.securityContext.appArm

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

- **type**

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

3.1.329. .spec.install.spec.deployments[].spec.template.spec.securityContext.seLinuxC

Description

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

Type

object

Property	Type	Description
level	**string**	Level is SELinux level label that applies to the container.
role	**string**	Role is a SELinux role label that applies to the container.
type	**string**	Type is a SELinux type label that applies to the container.
user	**string**	User is a SELinux user label that applies to the container.

3.1.330. .spec.install.spec.deployments[].spec.template.spec.securityContext.seccomp

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	**string**	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	**string**	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost – a profile defined in a file on the node should be used. RuntimeDefault – the container runtime default profile should be used. Unconfined – no profile should be applied.

3.1.331. .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. Note that this field cannot be set when spec.os.name is windows.

Type

array

3.1.332. .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
value	**string**	Value of a property to set

3.1.333. .spec.install.spec.deployments[].spec.template.spec.securityContext.windows

Description

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

Type

object

Property	Type	Description
gmsaCredentialSpec	**string**	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	**string**	GMSACredentialSpecName is the name of the GMSA credential spec to use.
hostProcess	**boolean**	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.

Property	Type	Description
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.

3.1.334. .spec.install.spec.deployments[].spec.template.spec.tolerations

Description

If specified, the pod's tolerations.

Type

array

3.1.335. .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.

Property	Type	Description
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.

3.1.336. .spec.install.spec.deployments[].spec.template.spec.topologySpreadConstrain

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

3.1.337. .spec.install.spec.deployments[].spec.template.spec.topologySpreadConstrain

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.
matchLabelKeys	**array (string)**	MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).

Property	Type	Description
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. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. | zone1 | zone2 | zone3 | | P P | P P | P | - if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1). - if MaxSkew is 2, incoming pod can be scheduled onto any zone. When **whenUnsatisfiable=Schedule Anyway**, it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.

Property	Type	Description
minDomains	**integer**	MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule. For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: | zone1 | zone2 | zone3 | | P P | P P | P P | The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.

Property	Type	Description
nodeAffinityPolicy	**string**	NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are: - Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. - Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations. If this value is nil, the behavior is equivalent to the Honor policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.
nodeTaintsPolicy	**string**	NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: - Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. - Ignore: node taints are ignored. All nodes are included. If this value is nil, the behavior is equivalent to the Ignore policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.

Property	Type	Description
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. We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of nodeAffinityPolicy and nodeTaintsPolicy. e.g. If TopologyKey is "kubernetes.io/hostname", each Node is a domain of that topology. And, if TopologyKey is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.

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 assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: | zone1 | zone2 | zone3 | | P P P | P | P | If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it **more** imbalanced. It's a required field.

3.1.338. .spec.install.spec.deployments[].spec.template.spec.topologySpreadConstrain

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.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Property	Type	Description
matchLabels	**object (string)**	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.

3.1.339. .spec.install.spec.deployments[].spec.template.spec.topologySpreadConstrain

Description

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

Type

array

3.1.340. .spec.install.spec.deployments[].spec.template.spec.topologySpreadConstrain

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 (string)**	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.

3.1.341. .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

3.1.342. .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 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
azureDisk	**object**	azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Property	Type	Description
azureFile	**object**	azureFile represents an Azure File Service mount on the host and bind mount to the pod.
cephfs	**object**	cephFS represents a Ceph FS mount on the host that shares a pod's lifetime
cinder	**object**	cinder represents a cinder volume attached and mounted on kubelets host machine. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
configMap	**object**	configMap represents a configMap that should populate this volume
csi	**object**	csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).
downwardAPI	**object**	downwardAPI represents downward API about the pod that should populate this volume
emptyDir	**object**	emptyDir represents a temporary directory that shares a pod's lifetime. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Property	Type	Description
ephemeral	**object**	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.
fc	**object**	fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.
flexVolume	**object**	flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Property	Type	Description
flocker	**object**	flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running
gcePersistentDisk	**object**	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/conc epts/storage/volumes#gcepersis tentdisk
gitRepo	**object**	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.
glusterfs	**object**	glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: https://examples.k8s.io/volumes/ glusterfs/README.md
hostPath	**object**	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/conc epts/storage/volumes#hostpath
image	**object**	image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided: – Always: the kubelet always

Property	Type	Description
		attempts to pull the reference. Container creation will fail If the pull fails. – Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. – IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails.
		The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (spec.containers[**].volumeMount s.mountPath) by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (spec.containers[**].**volumeMount s.subpath). The field spec.securityContext.fsGroupCha ngePolicy has no effect on this volume type.
iscsi	**object**	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

Property	Type	Description
name	**string**	name of the volume. Must be a DNS_LABEL and unique within the pod. More info: [https://kubernetes.io/docs/conc epts/overview/working-with-objects/names/#names](https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names)
nfs	**object**	nfs represents an NFS mount on the host that shares a pod's lifetime More info: [https://kubernetes.io/docs/conc epts/storage/volumes#nfs](https://kubernetes.io/docs/concepts/storage/volumes#nfs)
persistentVolumeClaim	**object**	persistentVolumeClaimVolumeSo urce represents a reference to a PersistentVolumeClaim in the same namespace. More info: [https://kubernetes.io/docs/conc epts/storage/persistent-volumes#persistentvolumeclaims](https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims)
photonPersistentDisk	**object**	photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine
portworxVolume	**object**	portworxVolume represents a portworx volume attached and mounted on kubelets host machine
projected	**object**	projected items for all in one resources secrets, configmaps, and downward API
quobyte	**object**	quobyte represents a Quobyte mount on the host that shares a pod's lifetime
rbd	**object**	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](https://examples.k8s.io/volumes/rbd/README.md)

Property	Type	Description
scaleIO	**object**	scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.
secret	**object**	secret represents a secret that should populate this volume. More info: [https://kubernetes.io/docs/conc epts/storage/volumes#secret](https://kubernetes.io/docs/concepts/storage/volumes#secret)
storageos	**object**	storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.
vsphereVolume	**object**	vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

3.1.343. .spec.install.spec.deployments[].spec.template.spec.volumes[].awsElasticBloc

Description

awsElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Type

object

Required

- **volumeID**

Property	Type	Description
fsType	**string**	fsType is the filesystem type of the volume that you want to 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/conc epts/storage/volumes#awselastic blockstore](https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore)

Property	Type	Description
partition	**integer**	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
readOnly	**boolean**	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	**string**	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

3.1.344. .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**	cachingMode is the Host Caching mode: None, Read Only, Read Write.
diskName	**string**	diskName is the Name of the data disk in the blob storage
diskURI	**string**	diskURI is the URI of data disk in the blob storage

Property	Type	Description
fsType	**string**	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
kind	**string**	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared
readOnly	**boolean**	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

3.1.345. .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**	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretName	**string**	secretName is the name of secret that contains Azure Storage Account Name and Key
shareName	**string**	shareName is the azure share Name

3.1.346. .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 (string)**	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	**string**	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
readOnly	**boolean**	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretFile	**string**	secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	**object**	secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

Property	Type	Description
user	**string**	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

3.1.347. .spec.install.spec.deployments[].spec.template.spec.volumes[].cephfs.secretR

Description

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

Type

object

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

3.1.348. .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

Property	Type	Description
fsType	**string**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
readOnly	**boolean**	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	**object**	secretRef is optional: points to a secret object containing parameters used to connect to OpenStack.
volumeID	**string**	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md

3.1.349. .spec.install.spec.deployments[].spec.template.spec.volumes[].cinder.secretR

Description

secretRef is optional: points to a secret object containing parameters used to connect to OpenStack.

Type

object

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

3.1.350. .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**	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	**array**	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified 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 '..'.
items[]	**object**	Maps a string key to a path within a volume.
name	**string**	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
optional	**boolean**	optional specify whether the ConfigMap or its keys must be defined

3.1.351. .spec.install.spec.deployments[].spec.template.spec.volumes[].configMap.item

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

3.1.352. .spec.install.spec.deployments[].spec.template.spec.volumes[].configMap.iter

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**

- **path**

Property	Type	Description
key	**string**	key is the key to project.
mode	**integer**	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
path	**string**	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

3.1.353. .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**	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
nodePublishSecretRef	**object**	nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.
readOnly	**boolean**	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).

Property	Type	Description
volumeAttributes	object (string)	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

3.1.354. .spec.install.spec.deployments[].spec.template.spec.volumes[].csi.nodePublis

Description

nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

object

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

3.1.355. .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

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
items[]	**object**	DownwardAPIVolumeFile represents information to create the file containing the pod field

3.1.356. .spec.install.spec.deployments[].spec.template.spec.volumes[].downwardAPI.i

Description

Items is a list of downward API volume file

Type

array

3.1.357. .spec.install.spec.deployments[].spec.template.spec.volumes[].downwardAPI.i

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

- **path**

Property	Type	Description

Property	Type	Description
fieldRef	**object**	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
mode	**integer**	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	**object**	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

3.1.358. .spec.install.spec.deployments[].spec.template.spec.volumes[].downwardAPI.i

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

object

Required

- **fieldPath**

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

3.1.359. .spec.install.spec.deployments[].spec.template.spec.volumes[].downwardAPI.

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-or-string**	Specifies the output format of the exposed resources, defaults to "1"
resource	**string**	Required: resource to select

3.1.360. .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

Property	Type	Description
medium	**string**	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: [https://kubernetes.io/docs/conc epts/storage/volumes#emptydir](https://kubernetes.io/docs/concepts/storage/volumes#emptydir)
sizeLimit	**integer-or-string**	sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: [https://kubernetes.io/docs/conc epts/storage/volumes#emptydir](https://kubernetes.io/docs/concepts/storage/volumes#emptydir)

3.1.361. .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.

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.

3.1.362. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.vol

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	**object**	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.

3.1.363. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.vol

Description

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.

Type
  object

3.1.364. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.vol

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

Property	Type	Description
accessModes	**array (string)**	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**	dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Property	Type	Description
dataSourceRef	**object**	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Property	Type	Description
resources	**object**	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources
selector	**object**	selector is a label query over volumes to consider for binding.
storageClassName	**string**	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

Property	Type	Description
volumeAttributesClassName	**string**	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that no VolumeAttributesClass will be applied to the claim but it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be set by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).
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.

3.1.365. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volu

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

- **kind**

- **name**

Property	Type	Description
apiGroup	**string**	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	**string**	Kind is the type of resource being referenced
name	**string**	Name is the name of resource being referenced

3.1.366. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.vol

Description

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

- **kind**

- **name**

Property	Type	Description
apiGroup	**string**	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	**string**	Kind is the type of resource being referenced
name	**string**	Name is the name of resource being referenced
namespace	**string**	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

3.1.367. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volu

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

Type

object

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

3.1.368. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volu

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

3.1.369. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volu

Description

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

Type

array

3.1.370. .spec.install.spec.deployments[].spec.template.spec.volumes[].ephemeral.volu

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 (string)**	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.

3.1.371. .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**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
lun	**integer**	lun is Optional: FC target lun number
readOnly	**boolean**	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
targetWWNs	**array (string)**	targetWWNs is Optional: FC target worldwide names (WWNs)
wwids	**array (string)**	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

3.1.372. .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.

Property	Type	Description
fsType	**string**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
options	**object (string)**	options is Optional: this field holds extra command options if any.
readOnly	**boolean**	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	**object**	secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

3.1.373. .spec.install.spec.deployments[].spec.template.spec.volumes[].flexVolume.se

Description

secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Type

object

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

3.1.374. .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**	datasetName is Name of the dataset stored as metadata → name on the dataset for Flocker should be considered as deprecated
datasetUUID	**string**	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

3.1.375. .spec.install.spec.deployments[].spec.template.spec.volumes[].gcePersistentI

Description

gcePersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Type

object

Required

- **pdName**

Property	Type	Description
fsType	**string**	fsType is filesystem type of the volume that you want to mount. 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#gcepersistentdisk

Property	Type	Description
partition	**integer**	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
pdName	**string**	pdName is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	**boolean**	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

3.1.376. .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

Property	Type	Description
directory	**string**	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
repository	**string**	repository is the URL
revision	**string**	revision is the commit hash for the specified revision.

3.1.377. .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**	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	**string**	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

Property	Type	Description
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

3.1.378. .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

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

3.1.379. .spec.install.spec.deployments[].spec.template.spec.volumes[].image

Description

image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided:

- Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails.

- Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present.

- IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails.

The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (spec.containers[**].volumeMounts.mountPath) by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (spec.containers[**].volumeMounts.subpath). The field spec.securityContext.fsGroupChangePolicy has no effect on this volume type.**

Type

object

Property	Type	Description
pullPolicy	**string**	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.

Property	Type	Description
reference	**string**	Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: [https://kubernetes.io/docs/conc epts/containers/images](https://kubernetes.io/docs/concepts/containers/images) This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

3.1.380. .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**	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
chapAuthSession	**boolean**	chapAuthSession defines whether support iSCSI Session CHAP authentication

Property	Type	Description
fsType	**string**	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	**string**	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface <target portal>:<volume name> will be created for the connection.
iqn	**string**	iqn is the target iSCSI Qualified Name.
iscsiInterface	**string**	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
lun	**integer**	lun represents iSCSI Target Lun number.
portals	**array (string)**	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
readOnly	**boolean**	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
secretRef	**object**	secretRef is the CHAP Secret for iSCSI target and initiator authentication
targetPortal	**string**	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

3.1.381. .spec.install.spec.deployments[].spec.template.spec.volumes[].iscsi.secretRef

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

Type

object

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

3.1.382. .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/conc epts/storage/volumes#nfs](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/conc epts/storage/volumes#nfs](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/conc epts/storage/volumes#nfs](https://kubernetes.io/docs/concepts/storage/volumes#nfs)

3.1.383. .spec.install.spec.deployments[].spec.template.spec.volumes[].persistentVolu

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/conc epts/storage/persistent-volumes#persistentvolumeclaims](https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims)
readOnly	**boolean**	readOnly Will force the ReadOnly setting in VolumeMounts. Default false.

3.1.384. .spec.install.spec.deployments[].spec.template.spec.volumes[].photonPersiste

Description

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

Type

object

Required

- **pdID**

Property	Type	Description
fsType	**string**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
pdID	**string**	pdID is the ID that identifies Photon Controller persistent disk

3.1.385. .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**	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	**string**	volumeID uniquely identifies a Portworx volume

3.1.386. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected

Description

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

Type

    object

Property	Type	Description
defaultMode	**integer**	defaultMode are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
sources	**array**	sources is the list of volume projections. Each entry in this list handles one source.
sources[]	**object**	Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

3.1.387. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

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

Type

array

3.1.388. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

object

Property	Type	Description

Property	Type	Description
clusterTrustBundle	**object**	ClusterTrustBundle allows a pod to access the **.spec.trustBundle** field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.
configMap	**object**	configMap information about the configMap data to project
downwardAPI	**object**	downwardAPI information about the downwardAPI data to project
secret	**object**	secret information about the secret data to project
serviceAccountToken	**object**	serviceAccountToken is information about the serviceAccountToken data to project

3.1.389. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

ClusterTrustBundle allows a pod to access the **.spec.trustBundle** field of ClusterTrustBundle objects in an auto-updating file.
Alpha, gated by the ClusterTrustBundleProjection feature gate.

ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector.

Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem.

Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.

Type

object

Required

- **path**

Property	Type	Description
labelSelector	**object**	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".
name	**string**	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
optional	**boolean**	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
path	**string**	Relative path from the volume root to write the bundle.
signerName	**string**	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

3.1.390. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

3.1.391. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

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

Type

array

3.1.392. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

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 (string)**	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.

3.1.393. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

configMap information about the configMap data to project

Type

object

Property	Type	Description
items	**array**	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	**object**	Maps a string key to a path within a volume.

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

3.1.394. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

3.1.395. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	**string**	key is the key to project.

Property	Type	Description
mode	**integer**	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

3.1.396. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

downwardAPI information about the downwardAPI data to project

Type

  object

Property	Type	Description
items	**array**	Items is a list of DownwardAPIVolume file
items[]	**object**	DownwardAPIVolumeFile represents information to create the file containing the pod field

3.1.397. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

Items is a list of DownwardAPIVolume file

Type

  array

3.1.398. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

- **path**

Property	Type	Description
fieldRef	**object**	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
mode	**integer**	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	**object**	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

3.1.399. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

object

Required

- **fieldPath**

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

3.1.400. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

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-or-string**	Specifies the output format of the exposed resources, defaults to "1"
resource	**string**	Required: resource to select

3.1.401. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

secret information about the secret data to project

Type

object

Property	Type	Description
items	**array**	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	**object**	Maps a string key to a path within a volume.
name	**string**	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	**boolean**	optional field specify whether the Secret or its key must be defined

3.1.402. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

> items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

> **array**

3.1.403. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

> Maps a string key to a path within a volume.

Type

object

Required

- **key**

- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

3.1.404. .spec.install.spec.deployments[].spec.template.spec.volumes[].projected.sour

Description

serviceAccountToken is information about the serviceAccountToken data to project

Type

object

Required

- **path**

Property	Type	Description

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.

3.1.405. .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

Property	Type	Description
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.

3.1.406. .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

Property	Type	Description
fsType	**string**	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	**string**	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	**string**	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitors	**array (string)**	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	**string**	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	**boolean**	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	**object**	secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Property	Type	Description
user	**string**	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

3.1.407. .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. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.408. .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

Property	Type	Description
fsType	**string**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
gateway	**string**	gateway is the host address of the ScaleIO API Gateway.
protectionDomain	**string**	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
readOnly	**boolean**	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	**object**	secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.
sslEnabled	**boolean**	sslEnabled Flag enable/disable SSL communication with Gateway, default false
storageMode	**string**	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
storagePool	**string**	storagePool is the ScaleIO Storage Pool associated with the protection domain.
system	**string**	system is the name of the storage system as configured in ScaleIO.
volumeName	**string**	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

3.1.409. .spec.install.spec.deployments[].spec.template.spec.volumes[].scaleIO.secret

Description

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

object

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

3.1.410. .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**	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
items	**array**	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	**object**	Maps a string key to a path within a volume.
optional	**boolean**	optional field specify whether the Secret or its keys must be defined
secretName	**string**	secretName is the name of the secret in the pod's namespace to use. More info: [https://kubernetes.io/docs/conc epts/storage/volumes#secret](https://kubernetes.io/docs/concepts/storage/volumes#secret)

3.1.411. .spec.install.spec.deployments[].spec.template.spec.volumes[].secret.items

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

3.1.412. .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**	key is the key to project.
mode	**integer**	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

3.1.413. .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**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	**boolean**	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Property	Type	Description
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.

3.1.414. .spec.install.spec.deployments[].spec.template.spec.volumes[].storageos.secr

Description

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

object

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

3.1.415. .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**	fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
storagePolicyID	**string**	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
storagePolicyName	**string**	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
volumePath	**string**	volumePath is the path that identifies vSphere volume vmdk

3.1.416. .spec.install.spec.permissions

Description

Type

array

3.1.417. .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**	
rules[]	**object**	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.
serviceAccountName	**string**	

3.1.418. .spec.install.spec.permissions[].rules

Description

Type

    array

3.1.419. .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 (string)**	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. "" represents the core API group and "*" represents all API groups.

Property	Type	Description
nonResourceURLs	**array (string)**	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 (string)**	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
resources	**array (string)**	Resources is a list of resources this rule applies to. '*' represents all resources.
verbs	**array (string)**	Verbs is a list of Verbs that apply to ALL the ResourceKinds contained in this rule. '*' represents all verbs.

3.1.420. .spec.installModes

Description

InstallModes specify supported installation types

Type

array

3.1.421. .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

3.1.422. .spec.links

Description

A list of links related to the operator.

Type

array

3.1.423. .spec.links[]

Description

Type

object

Property	Type	Description
name	**string**	
url	**string**	

3.1.424. .spec.maintainers

Description

A list of organizational entities maintaining the operator.

Type

array

3.1.425. .spec.maintainers[]

Description

Type

object

Property	Type	Description
email	**string**	

Property	Type	Description
name	**string**	

3.1.426. .spec.nativeAPIs

Description

Type

array

3.1.427. .spec.nativeAPIs[]

Description

GroupVersionKind unambiguously identifies a kind. It doesn't anonymously include GroupVersion to avoid automatic coercion. 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**	

3.1.428. .spec.provider

Description

The publishing entity behind the operator.

Type

object

Property	Type	Description
name	**string**	
url	**string**	

3.1.429. .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

3.1.430. .spec.relatedImages[]

Description

Type

object

Required

- **image**
- **name**

Property	Type	Description
image	**string**	
name	**string**	

3.1.431. .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.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Property	Type	Description
matchLabels	**object (string)**	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.

3.1.432. .spec.selector.matchExpressions

Description

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

Type

array

3.1.433. .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 (string)**	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.

3.1.434. .spec.webhookdefinitions

Description

Type

    array

3.1.435. .spec.webhookdefinitions[]

Description

    WebhookDescription provides details to OLM about required webhooks

Type

    object

Required

- **admissionReviewVersions**

- **generateName**

- **sideEffects**

- **type**

Property	Type	Description
admissionReviewVersions	**array (string)**	
containerPort	**integer**	
conversionCRDs	**array (string)**	
deploymentName	**string**	

Property	Type	Description
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**	
rules[]	**object**	RuleWithOperations is a tuple of Operations and Resources. It is recommended to make sure that all the tuple expansions are valid.
sideEffects	**string**	SideEffectClass specifies the types of side effects a webhook may have.
targetPort	**integer-or-string**	
timeoutSeconds	**integer**	
type	**string**	WebhookAdmissionType is the type of admission webhooks supported by OLM
webhookPath	**string**	

3.1.436. .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.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

3.1.437. .spec.webhookdefinitions[].objectSelector.matchExpressions

Description

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

Type

array

3.1.438. .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 (string)**	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.

3.1.439. .spec.webhookdefinitions[].rules

Description

Type

array

3.1.440. .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 (string)**	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 (string)**	APIVersions is the API versions the resources belong to. " **is all versions. If** " is present, the length of the slice must be one. Required.

Property	Type	Description
operations	**array (string)**	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 (string)**	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 that there are no scope restrictions. Subresources match the scope of their parent resource. Default is "*".

3.1.441. .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
conditions[]	**object**	Conditions appear in the status as a record of state transitions on the ClusterServiceVersion
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 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
requirementStatus[]	**object**	

3.1.442. .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.
pendingDeletion[]	**object**	ResourceList represents a list of resources which are of the same Group/Kind

3.1.443. .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

3.1.444. .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**	

Property	Type	Description
instances[]	**object**	
kind	**string**	

3.1.445. .status.cleanup.pendingDeletion[].instances

Description

Type

array

3.1.446. .status.cleanup.pendingDeletion[].instances[]

Description

Type

object

Required

- **name**

Property	Type	Description
name	**string**	
namespace	**string**	Namespace can be empty for cluster-scoped resources

3.1.447. .status.conditions

Description

List of conditions, a history of state transitions

Type

array

3.1.448. .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
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'

3.1.449. .status.requirementStatus

Description

The status of each requirement for this CSV

Type

array

3.1.450. .status.requirementStatus[]

Description

Type

object

Required

- **group**

- **kind**

- **message**

- **name**

- **status**

- **version**

Property	Type	Description
dependents	**array**	
dependents[]	**object**	DependentStatus is the status for a dependent requirement (to prevent infinite nesting)
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**	

3.1.451. .status.requirementStatus[].dependents

Description

Type

  array

3.1.452. .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**	
kind	**string**	
message	**string**	
status	**string**	StatusReason is a camelcased reason for the status of a RequirementStatus or DependentStatus
uuid	**string**	
version	**string**	

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/operators.coreos.com/v1alpha1/clusterserviceversions**

  - **GET**: list objects of kind ClusterServiceVersion

- **/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterserviceversions**

  - **DELETE**: delete collection of ClusterServiceVersion

  - **GET**: list objects of kind ClusterServiceVersion

  - **POST**: create a ClusterServiceVersion

- **/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterserviceversions/{name}**

  - **DELETE**: delete a 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

3.2.1. /apis/operators.coreos.com/v1alpha1/clusterserviceversions

HTTP method

GET

Description

list objects of kind ClusterServiceVersion

Table 3.1. HTTP responses

HTTP code	Reponse body
200 – OK	**ClusterServiceVersionList** schema
401 – Unauthorized	Empty

3.2.2. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterservice

HTTP method

DELETE

Description

delete collection of ClusterServiceVersion

Table 3.2. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ClusterServiceVersion

Table 3.3. HTTP responses

HTTP code	Reponse body
200 – OK	**ClusterServiceVersionList** schema
401 – Unauthorized	Empty

HTTP method

POST

Description

create a ClusterServiceVersion

Table 3.4. 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.

Table 3.5. Body parameters

Parameter	Type	Description
body	**ClusterServiceVersion** schema	

Table 3.6. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterServiceVersion** schema
201 - Created	**ClusterServiceVersion** schema
202 - Accepted	**ClusterServiceVersion** schema
401 - Unauthorized	Empty

3.2.3. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterservice

Table 3.7. Global path parameters

Parameter	Type	Description
name	**string**	name of the ClusterServiceVersion

HTTP method
> **DELETE**

Description
> delete a ClusterServiceVersion

Table 3.8. 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

Table 3.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method
> **GET**

Description
> read the specified ClusterServiceVersion

Table 3.10. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterServiceVersion** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ClusterServiceVersion

Table 3.11. 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.

Table 3.12. HTTP responses

HTTP code	Reponse body
200 – OK	**ClusterServiceVersion** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ClusterServiceVersion

Table 3.13. 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.

Table 3.14. Body parameters

Parameter	Type	Description
body	**ClusterServiceVersion** schema	

Table 3.15. HTTP responses

HTTP code	Reponse body
200 – OK	**ClusterServiceVersion** schema
201 – Created	**ClusterServiceVersion** schema
401 – Unauthorized	Empty

3.2.4. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/clusterservice

Table 3.16. Global path parameters

Parameter	Type	Description
name	**string**	name of the ClusterServiceVersion

HTTP method

GET

Description

read status of the specified ClusterServiceVersion

Table 3.17. HTTP responses

HTTP code	Reponse body
200 – OK	**ClusterServiceVersion** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ClusterServiceVersion

Table 3.18. 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.

Table 3.19. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterServiceVersion** schema
401 – Unauthorized	Empty

HTTP method

    PUT

Description

    replace status of the specified ClusterServiceVersion

Table 3.20. 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.

Table 3.21. Body parameters

Parameter	Type	Description
body	**ClusterServiceVersion** schema	

Table 3.22. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterServiceVersion** schema
201 – Created	**ClusterServiceVersion** schema
401 – Unauthorized	Empty

CHAPTER 4. INSTALLPLAN [OPERATORS.COREOS.COM/V1ALPHA1]

Description

InstallPlan defines the installation of a set of operators.

Type

object

Required

- **metadata**

- **spec**

4.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	InstallPlanSpec defines a set of Application resources to be installed

Property	Type	Description
status	**object**	InstallPlanStatus represents the information about the status of steps required to complete installation. Status may trail the actual state of a system.

4.1.1. .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 (string)**	
generation	**integer**	
source	**string**	
sourceNamespace	**string**	

4.1.2. .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.
bundleLookups[]	**object**	BundleLookup is a request to pull and unpackage the content of a bundle to the cluster.
catalogSources	**array (string)**	
conditions	**array**	
conditions[]	**object**	InstallPlanCondition represents the overall status of the execution of an InstallPlan.
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**	
plan[]	**object**	Step represents the status of an individual step in an InstallPlan.
startTime	**string**	StartTime is the time when the controller began applying the resources listed in the plan to the cluster.

4.1.3. .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.
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

4.1.4. .status.bundleLookups

Description

BundleLookups is the set of in-progress requests to pull and unpackage bundle content to the cluster.

Type

array

4.1.5. .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.

Property	Type	Description
conditions[]	**object**	

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.

4.1.6. .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.

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.
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: 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

4.1.7. .status.bundleLookups[].conditions

Description

Conditions represents the overall state of a BundleLookup.

Type

array

4.1.8. .status.bundleLookups[].conditions[]

Description

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.

4.1.9. .status.conditions

Description

Type

array

4.1.10. .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.

4.1.11. .status.plan

Description

Type

array

4.1.12. .status.plan[]

Description

Step represents the status of an individual step in an InstallPlan.

Type

object

Required

- **resolving**

- **resource**

- **status**

Property	Type	Description
optional	**boolean**	

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

4.1.13. .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**	

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/operators.coreos.com/v1alpha1/installplans**

  - **GET**: list objects of kind InstallPlan

- **/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans**

  - **DELETE**: delete collection of InstallPlan

  - **GET**: list objects of kind InstallPlan

  - **POST**: create an InstallPlan

- **/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans/{name}**

  - **DELETE**: delete an 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

4.2.1. /apis/operators.coreos.com/v1alpha1/installplans

HTTP method
    GET
Description
    list objects of kind InstallPlan

Table 4.1. HTTP responses

HTTP code	Reponse body
200 - OK	**InstallPlanList** schema
401 - Unauthorized	Empty

4.2.2. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans

HTTP method
    DELETE

Description

delete collection of InstallPlan

Table 4.2. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind InstallPlan

Table 4.3. HTTP responses

HTTP code	Reponse body
200 - OK	**InstallPlanList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an InstallPlan

Table 4.4. 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.

Table 4.5. Body parameters

Parameter	Type	Description
body	**InstallPlan** schema	

Table 4.6. HTTP responses

HTTP code	Reponse body
200 – OK	**InstallPlan** schema
201 – Created	**InstallPlan** schema
202 – Accepted	**InstallPlan** schema
401 – Unauthorized	Empty

4.2.3. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans/{r

Table 4.7. Global path parameters

Parameter	Type	Description
name	**string**	name of the InstallPlan

HTTP method

> **DELETE**

Description

> delete an InstallPlan

Table 4.8. 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

Table 4.9. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method

> **GET**

Description

> read the specified InstallPlan

Table 4.10. HTTP responses

HTTP code	Reponse body
200 – OK	**InstallPlan** schema
401 – Unauthorized	Empty

HTTP method

> **PATCH**

Description

> partially update the specified InstallPlan

Table 4.11. 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.

Table 4.12. HTTP responses

HTTP code	Reponse body
200 – OK	**InstallPlan** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace the specified InstallPlan

Table 4.13. Query parameters

Parameter	Type	Description

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.

Table 4.14. Body parameters

Parameter	Type	Description
body	**InstallPlan** schema	

Table 4.15. HTTP responses

HTTP code	Reponse body
200 – OK	**InstallPlan** schema
201 – Created	**InstallPlan** schema
401 – Unauthorized	Empty

4.2.4. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/installplans/{n

Table 4.16. Global path parameters

Parameter	Type	Description
name	**string**	name of the InstallPlan

HTTP method

> **GET**

Description

> read status of the specified InstallPlan

Table 4.17. HTTP responses

HTTP code	Reponse body
200 - OK	**InstallPlan** schema
401 – Unauthorized	Empty

HTTP method

> **PATCH**

Description

> partially update status of the specified InstallPlan

Table 4.18. 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.

Table 4.19. HTTP responses

HTTP code	Reponse body
200 – OK	**InstallPlan** schema
401 – Unauthorized	Empty

HTTP method

    PUT

Description

    replace status of the specified InstallPlan

Table 4.20. 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.

Table 4.21. Body parameters

Parameter	Type	Description
body	**InstallPlan** schema	

Table 4.22. HTTP responses

HTTP code	Reponse body
200 – OK	**InstallPlan** schema
201 – Created	**InstallPlan** schema
401 – Unauthorized	Empty

CHAPTER 5. OLMCONFIG [OPERATORS.COREOS.COM/V1]

Description

OLMConfig is a resource responsible for configuring OLM.

Type

object

Required

- **metadata**

5.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	OLMConfigSpec is the spec for an OLMConfig resource.
status	**object**	OLMConfigStatus is the status for an OLMConfig resource.

5.1.1. .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.

5.1.2. .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.
packageServerSyncInterval	**string**	PackageServerSyncInterval is used to define the sync interval for packagerserver pods. Packageserver pods periodically check the status of CatalogSources; this specifies the period using duration format (e.g. "60m"). For this parameter, only hours ("h"), minutes ("m"), and seconds ("s") may be specified. When not specified, the period defaults to the value specified within the packageserver.

5.1.3. .status

Description

OLMConfigStatus is the status for an OLMConfig resource.

Type

object

Property	Type	Description
conditions	**array**	
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.

5.1.4. .status.conditions

Description

Type

array

5.1.5. .status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource.

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.

Property	Type	Description
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.
type	**string**	type of condition in CamelCase or in foo.example.com/CamelCase.

5.2. 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 an OLMConfig

- **/apis/operators.coreos.com/v1/olmconfigs/{name}**

    - **DELETE**: delete an 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

5.2.1. /apis/operators.coreos.com/v1/olmconfigs

HTTP method

DELETE

Description

delete collection of OLMConfig

Table 5.1. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

list objects of kind OLMConfig

Table 5.2. HTTP responses

HTTP code	Reponse body
200 – OK	**OLMConfigList** schema
401 – Unauthorized	Empty

HTTP method

POST

Description

create an OLMConfig

Table 5.3. Query parameters

Parameter	Type	Description

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.

Table 5.4. Body parameters

Parameter	Type	Description
body	**OLMConfig** schema	

Table 5.5. HTTP responses

HTTP code	Reponse body
200 – OK	**OLMConfig** schema
201 – Created	**OLMConfig** schema
202 – Accepted	**OLMConfig** schema
401 – Unauthorized	Empty

5.2.2. /apis/operators.coreos.com/v1/olmconfigs/{name}

Table 5.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the OLMConfig

HTTP method

    DELETE

Description

    delete an OLMConfig

Table 5.7. 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

Table 5.8. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method

    GET

Description

    read the specified OLMConfig

Table 5.9. HTTP responses

HTTP code	Reponse body
200 – OK	**OLMConfig** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified OLMConfig

Table 5.10. 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.

Table 5.11. HTTP responses

HTTP code	Reponse body
200 – OK	**OLMConfig** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace the specified OLMConfig

Table 5.12. 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.

Table 5.13. Body parameters

Parameter	Type	Description
body	**OLMConfig** schema	

Table 5.14. HTTP responses

HTTP code	Reponse body
200 – OK	**OLMConfig** schema
201 – Created	**OLMConfig** schema
401 – Unauthorized	Empty

5.2.3. /apis/operators.coreos.com/v1/olmconfigs/{name}/status

Table 5.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the OLMConfig

HTTP method

GET

Description

read status of the specified OLMConfig

Table 5.16. HTTP responses

HTTP code	Reponse body
200 – OK	**OLMConfig** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified OLMConfig

Table 5.17. 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.

Table 5.18. HTTP responses

HTTP code	Reponse body
200 – OK	**OLMConfig** schema
401 – Unauthorized	Empty

HTTP method

   PUT

Description

   replace status of the specified OLMConfig

Table 5.19. 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.

Table 5.20. Body parameters

Parameter	Type	Description
body	**OLMConfig** schema	

Table 5.21. HTTP responses

HTTP code	Reponse body
200 – OK	**OLMConfig** schema
201 – Created	**OLMConfig** schema
401 – Unauthorized	Empty

CHAPTER 6. OPERATOR [OPERATORS.COREOS.COM/V1]

Description

Operator represents a cluster operator.

Type

object

6.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	OperatorSpec defines the desired state of Operator
status	**object**	OperatorStatus defines the observed state of an Operator and its components

6.1.1. .spec

Description

OperatorSpec defines the desired state of Operator

Type

object

6.1.2. .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.

6.1.3. .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.
refs[]	**object**	RichReference is a reference to a resource, enriched with its status conditions.

6.1.4. .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.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

6.1.5. .status.components.labelSelector.matchExpressions

Description

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

Type

array

6.1.6. .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.

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 (string)**	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.

6.1.7. .status.components.refs

Description

Refs are a set of references to the operator's component resources, selected with LabelSelector.

Type

array

6.1.8. .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.
conditions[]	**object**	Condition represent the latest available observations of an component's state.

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.
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: 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

6.1.9. .status.components.refs[].conditions

Description

Conditions represents the latest state of the component.

Type

array

6.1.10. .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.
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.

6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/operators.coreos.com/v1/operators**

  - **DELETE**: delete collection of Operator

  - **GET**: list objects of kind Operator

- **POST**: create an Operator

- **/apis/operators.coreos.com/v1/operators/{name}**

  - **DELETE**: delete an Operator

  - **GET**: read the specified Operator

  - **PATCH**: partially update the specified Operator

  - **PUT**: replace the specified Operator

- **/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

6.2.1. /apis/operators.coreos.com/v1/operators

HTTP method
   DELETE
Description
   delete collection of Operator

Table 6.1. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
401 – Unauthorized	Empty

HTTP method
   GET
Description
   list objects of kind Operator

Table 6.2. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorList** schema
401 – Unauthorized	Empty

HTTP method
   POST

Description

create an Operator

Table 6.3. 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.

Table 6.4. Body parameters

Parameter	Type	Description
body	**Operator** schema	

Table 6.5. HTTP responses

HTTP code	Reponse body
200 - OK	**Operator** schema
201 - Created	**Operator** schema
202 - Accepted	**Operator** schema
401 - Unauthorized	Empty

6.2.2. /apis/operators.coreos.com/v1/operators/{name}

Table 6.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Operator

HTTP method

DELETE

Description

delete an Operator

Table 6.7. 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

Table 6.8. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
202 - Accepted	**Status** schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Operator

Table 6.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Operator** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Operator

Table 6.10. 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.

Table 6.11. HTTP responses

HTTP code	Reponse body
200 – OK	**Operator** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Operator

Table 6.12. 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.

Table 6.13. Body parameters

Parameter	Type	Description
body	**Operator** schema	

Table 6.14. HTTP responses

HTTP code	Reponse body
200 – OK	**Operator** schema
201 – Created	**Operator** schema
401 – Unauthorized	Empty

6.2.3. /apis/operators.coreos.com/v1/operators/{name}/status

Table 6.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Operator

HTTP method

GET

Description

read status of the specified Operator

Table 6.16. HTTP responses

HTTP code	Reponse body
200 – OK	**Operator** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Operator

Table 6.17. 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.

Table 6.18. HTTP responses

HTTP code	Reponse body
200 – OK	**Operator** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Operator

Table 6.19. 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.

Table 6.20. Body parameters

Parameter	Type	Description
body	**Operator** schema	

Table 6.21. HTTP responses

HTTP code	Reponse body
200 – OK	**Operator** schema
201 – Created	**Operator** schema
401 – Unauthorized	Empty

CHAPTER 7. OPERATORCONDITION [OPERATORS.COREOS.COM/V2]

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**

7.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

Property	Type	Description
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.

7.1.1. .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**	
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.
deployments	**array (string)**	
overrides	**array**	
overrides[]	**object**	Condition contains details for one aspect of the current state of this API Resource.
serviceAccounts	**array (string)**	

7.1.2. .spec.conditions

Description

Type

array

7.1.3. .spec.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource.

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.
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.

Property	Type	Description
status	**string**	status of the condition, one of True, False, Unknown.
type	**string**	type of condition in CamelCase or in foo.example.com/CamelCase.

7.1.4. .spec.overrides

Description

Type

array

7.1.5. .spec.overrides[]

Description

Condition contains details for one aspect of the current state of this API Resource.

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.

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.

7.1.6. .status

Description

OperatorConditionStatus allows OLM to convey which conditions have been observed.

Type

object

Property	Type	Description
conditions	**array**	
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.

7.1.7. .status.conditions

Description

Type

array

7.1.8. .status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource.

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.

7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/operators.coreos.com/v2/operatorconditions**

    - **GET**: list objects of kind OperatorCondition

- **/apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions**

    - **DELETE**: delete collection of OperatorCondition

    - **GET**: list objects of kind OperatorCondition

    - **POST**: create an OperatorCondition

- **/apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions/{name}**

    - **DELETE**: delete an 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

7.2.1. /apis/operators.coreos.com/v2/operatorconditions

HTTP method
> **GET**

Description
> list objects of kind OperatorCondition

Table 7.1. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorConditionList** schema
401 - Unauthorized	Empty

7.2.2. /apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions

HTTP method
> **DELETE**

Description
> delete collection of OperatorCondition

Table 7.2. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
401 - Unauthorized	Empty

HTTP method
> **GET**

Description
> list objects of kind OperatorCondition

Table 7.3. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorConditionList** schema
401 - Unauthorized	Empty

HTTP method
> **POST**

Description

create an OperatorCondition

Table 7.4. 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.

Table 7.5. Body parameters

Parameter	Type	Description
body	**OperatorCondition** schema	

Table 7.6. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorCondition** schema
201 - Created	**OperatorCondition** schema
202 - Accepted	**OperatorCondition** schema
401 - Unauthorized	Empty

7.2.3. /apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions/

Table 7.7. Global path parameters

Parameter	Type	Description
name	**string**	name of the OperatorCondition

HTTP method
DELETE
Description
delete an OperatorCondition

Table 7.8. 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

Table 7.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method
GET
Description
read the specified OperatorCondition

Table 7.10. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorCondition** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified OperatorCondition

Table 7.11. 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.

Table 7.12. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorCondition** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace the specified OperatorCondition

Table 7.13. 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.

Table 7.14. Body parameters

Parameter	Type	Description
body	**OperatorCondition** schema	

Table 7.15. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorCondition** schema
201 – Created	**OperatorCondition** schema
401 – Unauthorized	Empty

7.2.4. /apis/operators.coreos.com/v2/namespaces/{namespace}/operatorconditions

Table 7.16. Global path parameters

Parameter	Type	Description
name	**string**	name of the OperatorCondition

HTTP method

    GET

Description

    read status of the specified OperatorCondition

Table 7.17. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorCondition** schema
401 – Unauthorized	Empty

HTTP method

    PATCH

Description

    partially update status of the specified OperatorCondition

Table 7.18. 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.

Table 7.19. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorCondition** schema
401 - Unauthorized	Empty

HTTP method

    PUT

Description

    replace status of the specified OperatorCondition

Table 7.20. 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.

Table 7.21. Body parameters

Parameter	Type	Description
body	**OperatorCondition** schema	

Table 7.22. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorCondition** schema
201 – Created	**OperatorCondition** schema
401 – Unauthorized	Empty

CHAPTER 8. 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**

8.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	OperatorGroupSpec is the spec for an OperatorGroup resource.

Property	Type	Description
status	**object**	OperatorGroupStatus is the status for an OperatorGroupResource.

8.1.1. .spec

Description

OperatorGroupSpec is the spec for an OperatorGroup resource.

Type

object

Property	Type	Description
selector	**object**	Selector selects the OperatorGroup's target namespaces.
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 (string)**	TargetNamespaces is an explicit set of namespaces to target. If it is set, Selector is ignored.

Property	Type	Description
upgradeStrategy	**string**	UpgradeStrategy defines the upgrade strategy for operators in the namespace. There are currently two supported upgrade strategies: Default: OLM will only allow clusterServiceVersions to move to the replacing phase from the succeeded phase. This effectively means that OLM will not allow operators to move to the next version if an installation or upgrade has failed. TechPreviewUnsafeFailForward: OLM will allow clusterServiceVersions to move to the replacing phase from the succeeded phase or from the failed phase. Additionally, OLM will generate new installPlans when a subscription references a failed installPlan and the catalog has been updated with a new upgrade for the existing set of operators. WARNING: The TechPreviewUnsafeFailForward upgrade strategy is unsafe and may result in unexpected behavior or unrecoverable data loss unless you have deep understanding of the set of operators being managed in the namespace.

8.1.2. .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.

Property	Type	Description
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

8.1.3. .spec.selector.matchExpressions

Description

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

Type

array

8.1.4. .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 (string)**	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.

8.1.5. .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.
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.
lastUpdated	**string**	LastUpdated is a timestamp of the last time the OperatorGroup's status was Updated.
namespaces	**array (string)**	Namespaces is the set of target namespaces for the OperatorGroup.
serviceAccountRef	**object**	ServiceAccountRef references the service account object specified.

8.1.6. .status.conditions

Description

Conditions is an array of the OperatorGroup's conditions.

Type

array

8.1.7. .status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource.

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.

8.1.8. .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.

Property	Type	Description
kind	**string**	Kind of the referent. More info: [https://git.k8s.io/community/con tributors/devel/sig-architecture/api-conventions.md#types-kinds](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/conc epts/overview/working-with-objects/names/#names](https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names)
namespace	**string**	Namespace of the referent. More info: [https://kubernetes.io/docs/conc epts/overview/working-with-objects/namespaces/](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/con tributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency](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/conc epts/overview/working-with-objects/names/#uids](https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids)

8.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/operators.coreos.com/v1/operatorgroups**

  - **GET**: list objects of kind OperatorGroup

- **/apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups**

  - **DELETE**: delete collection of OperatorGroup

  - **GET**: list objects of kind OperatorGroup

  - **POST**: create an OperatorGroup

- **/apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups/{name}**

  - **DELETE**: delete an 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

8.2.1. /apis/operators.coreos.com/v1/operatorgroups

HTTP method

GET

Description

list objects of kind OperatorGroup

Table 8.1. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorGroupList** schema
401 – Unauthorized	Empty

8.2.2. /apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups

HTTP method

DELETE

Description

delete collection of OperatorGroup

Table 8.2. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

list objects of kind OperatorGroup

Table 8.3. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorGroupList** schema
401 – Unauthorized	Empty

HTTP method

> **POST**

Description

> create an OperatorGroup

Table 8.4. 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.

Table 8.5. Body parameters

Parameter	Type	Description
body	**OperatorGroup** schema	

Table 8.6. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorGroup** schema
201 - Created	**OperatorGroup** schema
202 - Accepted	**OperatorGroup** schema
401 - Unauthorized	Empty

8.2.3. /apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups/{na

Table 8.7. Global path parameters

Parameter	Type	Description
name	**string**	name of the OperatorGroup

HTTP method
> **DELETE**

Description
> delete an OperatorGroup

Table 8.8. 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

Table 8.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
202 - Accepted	**Status** schema
401 - Unauthorized	Empty

HTTP method
> **GET**

Description

read the specified OperatorGroup

Table 8.10. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorGroup** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified OperatorGroup

Table 8.11. 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.

Table 8.12. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorGroup** schema
401 – Unauthorized	Empty

HTTP method

> **PUT**

Description

> replace the specified OperatorGroup

Table 8.13. 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.

Table 8.14. Body parameters

Parameter	Type	Description
body	**OperatorGroup** schema	

Table 8.15. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorGroup** schema
201 - Created	**OperatorGroup** schema
401 - Unauthorized	Empty

8.2.4. /apis/operators.coreos.com/v1/namespaces/{namespace}/operatorgroups/{na

Table 8.16. Global path parameters

Parameter	Type	Description
name	**string**	name of the OperatorGroup

HTTP method

    GET

Description

    read status of the specified OperatorGroup

Table 8.17. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorGroup** schema
401 - Unauthorized	Empty

HTTP method

    PATCH

Description

    partially update status of the specified OperatorGroup

Table 8.18. 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.

Table 8.19. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorGroup** schema
401 – Unauthorized	Empty

HTTP method

    PUT

Description

    replace status of the specified OperatorGroup

Table 8.20. 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.

Table 8.21. Body parameters

Parameter	Type	Description
body	**OperatorGroup** schema	

Table 8.22. HTTP responses

HTTP code	Reponse body
200 – OK	**OperatorGroup** schema
201 – Created	**OperatorGroup** schema
401 – Unauthorized	Empty

CHAPTER 9. PACKAGEMANIFEST [PACKAGES.OPERATORS.COREOS.COM/V1]

Description

PackageManifest holds information about a package, which is a reference to one (or more) channels under a single package.

Type

object

9.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	
spec	**object**	PackageManifestSpec defines the desired state of PackageManifest
status	**object**	PackageManifestStatus represents the current status of the PackageManifest

9.1.1. .spec

Description

PackageManifestSpec defines the desired state of PackageManifest

Type

object

9.1.2. .status

Description

PackageManifestStatus represents the current status of the PackageManifest

Type

object

Required

- **catalogSource**

- **catalogSourceDisplayName**

- **catalogSourcePublisher**

- **catalogSourceNamespace**

- **packageName**

- **channels**

- **defaultChannel**

Property	Type	Description
catalogSource	**string**	CatalogSource is the name of the CatalogSource this package belongs to
catalogSourceDisplayName	**string**	
catalogSourceNamespace	**string**	CatalogSourceNamespace is the namespace of the owning CatalogSource
catalogSourcePublisher	**string**	
channels	**array**	Channels are the declared channels for the package, ala **stable** or **alpha**.
channels[]	**object**	PackageChannel defines a single channel under a package, pointing to a version of that package.

Property	Type	Description
defaultChannel	**string**	DefaultChannel is, if specified, the name of the default channel for the package. The default channel will be installed if no other channel is explicitly given. If the package has a single channel, then that channel is implicitly the default.
deprecation	**object**	Deprecation conveys information regarding a deprecated resource.
packageName	**string**	PackageName is the name of the overall package, ala **etcd**.
provider	**object**	AppLink defines a link to an application

9.1.3. .status.channels

Description

Channels are the declared channels for the package, ala **stable** or **alpha**.

Type

    array

9.1.4. .status.channels[]

Description

PackageChannel defines a single channel under a package, pointing to a version of that package.

Type

    object

Required

- **name**
- **currentCSV**
- **entries**

Property	Type	Description
currentCSV	**string**	CurrentCSV defines a reference to the CSV holding the version of this package currently for the channel.

Property	Type	Description
currentCSVDesc	**object**	CSVDescription defines a description of a CSV
deprecation	**object**	Deprecation conveys information regarding a deprecated resource.
entries	**array**	Entries lists all CSVs in the channel, with their upgrade edges.
entries[]	**object**	ChannelEntry defines a member of a package channel.
name	**string**	Name is the name of the channel, e.g. **alpha** or **stable**

9.1.5. .status.channels[].currentCSVDesc

Description

CSVDescription defines a description of a CSV

Type

object

Property	Type	Description
annotations	**object (string)**	
apiservicedefinitions	**APIServiceDefinitions**	
customresourcedefinitions	**CustomResourceDefinitions**	
description	**string**	LongDescription is the CSV's description
displayName	**string**	DisplayName is the CSV's display name
icon	**array**	Icon is the CSV's base64 encoded icon
icon[]	**object**	Icon defines a base64 encoded icon and media type
installModes	**array (InstallMode)**	InstallModes specify supported installation types

Property	Type	Description
keywords	array (string)	
links	array	
links[]	object	AppLink defines a link to an application
maintainers	array	
maintainers[]	object	Maintainer defines a project maintainer
maturity	string	
minKubeVersion	string	Minimum Kubernetes version for operator installation
nativeApis	array (GroupVersionKind)	
provider	object	AppLink defines a link to an application
relatedImages	array (string)	List of related images
version	OperatorVersion	Version is the CSV's semantic version

9.1.6. .status.channels[].currentCSVDesc.icon

Description

Icon is the CSV's base64 encoded icon

Type

array

9.1.7. .status.channels[].currentCSVDesc.icon[]

Description

Icon defines a base64 encoded icon and media type

Type

object

Property	Type	Description
base64data	string	

Property	Type	Description
mediatype	string	

9.1.8. .status.channels[].currentCSVDesc.links

Description

Type

   array

9.1.9. .status.channels[].currentCSVDesc.links[]

Description

   AppLink defines a link to an application

Type

   object

Property	Type	Description
name	string	
url	string	

9.1.10. .status.channels[].currentCSVDesc.maintainers

Description

Type

   array

9.1.11. .status.channels[].currentCSVDesc.maintainers[]

Description

   Maintainer defines a project maintainer

Type

   object

Property	Type	Description
email	string	
name	string	

9.1.12. .status.channels[].currentCSVDesc.provider

Description

AppLink defines a link to an application

Type

object

Property	Type	Description
name	**string**	
url	**string**	

9.1.13. .status.channels[].deprecation

Description

Deprecation conveys information regarding a deprecated resource.

Type

object

Required

- **message**

Property	Type	Description
message	**string**	Message is a human readable message describing the deprecation.

9.1.14. .status.channels[].entries

Description

Entries lists all CSVs in the channel, with their upgrade edges.

Type

array

9.1.15. .status.channels[].entries[]

Description

ChannelEntry defines a member of a package channel.

Type

object

Required

- **name**

Property	Type	Description
deprecation	**object**	Deprecation conveys information regarding a deprecated resource.
name	**string**	Name is the name of the bundle for this entry.
version	**string**	Version is the version of the bundle for this entry.

9.1.16. .status.channels[].entries[].deprecation

Description

Deprecation conveys information regarding a deprecated resource.

Type

object

Required

- **message**

Property	Type	Description
message	**string**	Message is a human readable message describing the deprecation.

9.1.17. .status.deprecation

Description

Deprecation conveys information regarding a deprecated resource.

Type

object

Required

- **message**

Property	Type	Description
message	**string**	Message is a human readable message describing the deprecation.

9.1.18. .status.provider

Description

AppLink defines a link to an application

Type

object

Property	Type	Description
name	**string**	
url	**string**	

9.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/packages.operators.coreos.com/v1/packagemanifests**

  - **GET**: list objects of kind PackageManifest

- **/apis/packages.operators.coreos.com/v1/namespaces/{namespace}/packagemanifests**

  - **GET**: list objects of kind PackageManifest

- **/apis/packages.operators.coreos.com/v1/namespaces/{namespace}/packagemanifests/{name}**

  - **GET**: read the specified PackageManifest

- **/apis/packages.operators.coreos.com/v1/namespaces/{namespace}/packagemanifests/{name}/icon**

  - **GET**: connect GET requests to icon of PackageManifest

9.2.1. /apis/packages.operators.coreos.com/v1/packagemanifests

HTTP method

GET

Description

list objects of kind PackageManifest

Table 9.1. HTTP responses

HTTP code	Reponse body
200 - OK	**PackageManifestList** schema

9.2.2. /apis/packages.operators.coreos.com/v1/namespaces/{namespace}/packagem

HTTP method

GET

Description

list objects of kind PackageManifest

Table 9.2. HTTP responses

HTTP code	Reponse body
200 - OK	**PackageManifestList** schema

9.2.3. /apis/packages.operators.coreos.com/v1/namespaces/{namespace}/packagem

Table 9.3. Global path parameters

Parameter	Type	Description
name	**string**	name of the PackageManifest

HTTP method

GET

Description

read the specified PackageManifest

Table 9.4. HTTP responses

HTTP code	Reponse body
200 - OK	**PackageManifest** schema

9.2.4. /apis/packages.operators.coreos.com/v1/namespaces/{namespace}/packagem

Table 9.5. Global path parameters

Parameter	Type	Description
name	**string**	name of the PackageManifest

HTTP method

GET

Description

connect GET requests to icon of PackageManifest

Table 9.6. HTTP responses

HTTP code	Reponse body
200 - OK	**string**

CHAPTER 10. SUBSCRIPTION [OPERATORS.COREOS.COM/V1ALPHA1]

Description

Subscription keeps operators up to date by tracking changes to Catalogs.

Type

object

Required

- **metadata**

- **spec**

10.1. 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	SubscriptionSpec defines an Application that can be installed

Property	Type	Description
status	**object**	

10.1.1. .spec

Description

SubscriptionSpec defines an Application that can be installed

Type

object

Required

- **name**

- **source**

- **sourceNamespace**

Property	Type	Description
channel	**string**	
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**	

10.1.2. .spec.config

Description

SubscriptionConfig contains configuration specified for a subscription.

Type

object

Property	Type	Description
affinity	**object**	If specified, overrides the pod's scheduling constraints. nil sub-attributes will **not** override the original values in the pod.spec for those sub-attributes. Use empty object ({}) to erase original sub-attribute values.
annotations	**object (string)**	Annotations is an unstructured key value map stored with each Deployment, Pod, APIService in the Operator. Typically, annotations may be set by external tools to store and retrieve arbitrary metadata. Use this field to pre-define annotations that OLM should add to each of the Subscription's deployments, pods, and apiservices.
env	**array**	Env is a list of environment variables to set in the container. Cannot be updated.
env[]	**object**	EnvVar represents an environment variable present in a Container.
envFrom	**array**	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.
envFrom[]	**object**	EnvFromSource represents the source of a set of ConfigMaps

Property	Type	Description
nodeSelector	**object (string)**	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.
tolerations[]	**object**	The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.
volumeMounts	**array**	List of VolumeMounts to set in the container.
volumeMounts[]	**object**	VolumeMount describes a mounting of a Volume within a container.
volumes	**array**	List of Volumes to set in the podSpec.
volumes[]	**object**	Volume represents a named volume in a pod that may be accessed by any container in the pod.

10.1.3. .spec.config.affinity

Description

If specified, overrides the pod's scheduling constraints. nil sub-attributes will **not** override the original values in the pod.spec for those sub-attributes. Use empty object ({}) to erase original sub-attribute values.

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)).

10.1.4. .spec.config.affinity.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description

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

10.1.5. .spec.config.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExe

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

10.1.6. .spec.config.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExe

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.

10.1.7. .spec.config.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExe

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.
matchExpressions[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	**array**	A list of node selector requirements by node's fields.

Property	Type	Description
matchFields[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

10.1.8. .spec.config.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExe

Description

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

Type

array

10.1.9. .spec.config.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExe

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 (string)**	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.

10.1.10. .spec.config.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExe

Description

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

Type

array

10.1.11. .spec.config.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExe

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 (string)**	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.

10.1.12. .spec.config.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExec

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.
nodeSelectorTerms[]	**object**	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.

10.1.13. .spec.config.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExec

Description

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

Type

array

10.1.14. .spec.config.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExec

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.
matchExpressions[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	**array**	A list of node selector requirements by node's fields.
matchFields[]	**object**	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

10.1.15. .spec.config.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExec

Description

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

Type

array

10.1.16. .spec.config.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExec

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

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 (string)**	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.

10.1.17. .spec.config.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExec

Description

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

Type

array

10.1.18. .spec.config.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExec

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 (string)**	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.

10.1.19. .spec.config.affinity.podAffinity

Description

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

Type

object

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

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution[]	**object**	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	**array**	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to 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.

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution[]	**object**	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

10.1.20. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExe

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

10.1.21. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExec

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.

Property	Type	Description
weight	**integer**	weight associated with matching the corresponding podAffinityTerm, in the range 1–100.

10.1.22. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExe

Description

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

Type

object

Required

- **topologyKey**

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

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

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

10.1.23. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExe

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.24. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExe

Description

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

Type

array

10.1.25. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExe

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 (string)**	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.

10.1.26. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExe

Description

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

Type

object

Property	Type	Description

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.27. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExe

Description

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

Type

array

10.1.28. .spec.config.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExe

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 (string)**	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.

10.1.29. .spec.config.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExec

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

10.1.30. .spec.config.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExec

Description

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

Type

object

Required

- **topologyKey**

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

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

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

10.1.31. .spec.config.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecu

Description

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

Type

> **object**

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.32. .spec.config.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExec

Description

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

Type

array

10.1.33. .spec.config.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExec

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 (string)**	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.

10.1.34. .spec.config.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExec

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.35. .spec.config.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExec

Description

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

Type

array

10.1.36. .spec.config.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExec

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 (string)**	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.

10.1.37. .spec.config.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

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	**array**	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.
preferredDuringSchedulingIgnoredDuringExecution[]	**object**	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	**array**	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.

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution[]	**object**	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

10.1.38. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

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

10.1.39. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

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.

Property	Type	Description
weight	**integer**	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

10.1.40. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

Description

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

Type

object

Required

- **topologyKey**

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

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

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

10.1.41. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.42. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

Description

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

Type

array

10.1.43. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

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 (string)**	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.

10.1.44. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

Description

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

Type

object

Property	Type	Description

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.45. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

Description

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

Type

array

10.1.46. .spec.config.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

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 (string)**	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.

10.1.47. .spec.config.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringE

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

10.1.48. .spec.config.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringE

Description

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

Type

object

Required

- **topologyKey**

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

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

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

10.1.49. .spec.config.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringI

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.50. .spec.config.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringE

Description

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

Type

array

10.1.51. .spec.config.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringE

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 (string)**	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.

10.1.52. .spec.config.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringE

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.53. .spec.config.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringE

Description

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

Type

array

10.1.54. .spec.config.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringE

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 (string)**	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.

10.1.55. .spec.config.env

Description

Env is a list of environment variables to set in the container. Cannot be updated.

Type

array

10.1.56. .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 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.

10.1.57. .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

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

10.1.59. .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.

10.1.60. .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-or-string**	Specifies the output format of the exposed resources, defaults to "1"
resource	**string**	Required: resource to select

10.1.61. .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.
name	**string**	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	**boolean**	Specify whether the Secret or its key must be defined

10.1.62. .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

10.1.63. .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

10.1.64. .spec.config.envFrom[].configMapRef

Description

The ConfigMap to select from

Type

object

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

10.1.65. .spec.config.envFrom[].secretRef

Description

The Secret to select from

Type

object

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

10.1.66. .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
claims	**array**	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	**object**	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	**integer-or-string**	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	**integer-or-string**	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

10.1.67. .spec.config.resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type
array

10.1.68. .spec.config.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

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

10.1.69. .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.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Property	Type	Description
matchLabels	**object (string)**	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.

10.1.70. .spec.config.selector.matchExpressions

Description

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

Type

array

10.1.71. .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.

Property	Type	Description
values	array (string)	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.

10.1.72. .spec.config.tolerations

Description

Tolerations are the pod's tolerations.

Type

array

10.1.73. .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, 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.

Property	Type	Description
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.

10.1.74. .spec.config.volumeMounts

Description

List of VolumeMounts to set in the container.

Type

array

10.1.75. .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. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).
name	**string**	This must match the Name of a Volume.
readOnly	**boolean**	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	**string**	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	**string**	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPathExpr	**string**	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references $(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

10.1.76. .spec.config.volumes

Description

List of Volumes to set in the podSpec.

Type

array

10.1.77. .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 resource that is attached to a kubelet's host machine and then exposed to the pod. More info: [https://kubernetes.io/docs/conc epts/storage/volumes#awselastic blockstore](https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore)
azureDisk	**object**	azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.
azureFile	**object**	azureFile represents an Azure File Service mount on the host and bind mount to the pod.
cephfs	**object**	cephFS represents a Ceph FS mount on the host that shares a pod's lifetime
cinder	**object**	cinder represents a cinder volume attached and mounted on kubelets host machine. More info: [https://examples.k8s.io/mysql- cinder-pd/README.md](https://examples.k8s.io/mysql-cinder-pd/README.md)
configMap	**object**	configMap represents a configMap that should populate this volume
csi	**object**	csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Property	Type	Description
downwardAPI	**object**	downwardAPI represents downward API about the pod that should populate this volume
emptyDir	**object**	emptyDir represents a temporary directory that shares a pod's lifetime. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Property	Type	Description
ephemeral	**object**	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.
fc	**object**	fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.
flexVolume	**object**	flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Property	Type	Description
flocker	**object**	flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running
gcePersistentDisk	**object**	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/conc epts/storage/volumes#gcepersis tentdisk](https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk)
gitRepo	**object**	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.
glusterfs	**object**	glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: [https://examples.k8s.io/volumes/ glusterfs/README.md](https://examples.k8s.io/volumes/glusterfs/README.md)
hostPath	**object**	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/conc epts/storage/volumes#hostpath](https://kubernetes.io/docs/concepts/storage/volumes#hostpath)
image	**object**	image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided: - Always: the kubelet always

Property	Type	Description
		attempts to pull the reference. Container creation will fail If the pull fails. - Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. - IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (spec.containers[**].volumeMounts.mountPath) by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (spec.containers[**].volumeMount**s.subpath). The field spec.securityContext.fsGroupChangePolicy has no effect on this volume type.
iscsi	**object**	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

Property	Type	Description
name	**string**	name of the volume. Must be a DNS_LABEL and unique within the pod. More info: [https://kubernetes.io/docs/conc epts/overview/working-with-objects/names/#names](https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names)
nfs	**object**	nfs represents an NFS mount on the host that shares a pod's lifetime More info: [https://kubernetes.io/docs/conc epts/storage/volumes#nfs](https://kubernetes.io/docs/concepts/storage/volumes#nfs)
persistentVolumeClaim	**object**	persistentVolumeClaimVolumeSo urce represents a reference to a PersistentVolumeClaim in the same namespace. More info: [https://kubernetes.io/docs/conc epts/storage/persistent-volumes#persistentvolumeclaims](https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims)
photonPersistentDisk	**object**	photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine
portworxVolume	**object**	portworxVolume represents a portworx volume attached and mounted on kubelets host machine
projected	**object**	projected items for all in one resources secrets, configmaps, and downward API
quobyte	**object**	quobyte represents a Quobyte mount on the host that shares a pod's lifetime
rbd	**object**	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
scaleIO	**object**	scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Property	Type	Description
secret	**object**	secret represents a secret that should populate this volume. More info: [https://kubernetes.io/docs/conc epts/storage/volumes#secret](https://kubernetes.io/docs/concepts/storage/volumes#secret)
storageos	**object**	storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.
vsphereVolume	**object**	vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

10.1.78. .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**	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: [https://kubernetes.io/docs/conc epts/storage/volumes#awselastic blockstore](https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore)

Property	Type	Description
partition	**integer**	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
readOnly	**boolean**	readOnly value true will force the readOnly setting in VolumeMounts. More info: [https://kubernetes.io/docs/conc epts/storage/volumes#awselastic blockstore](https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore)
volumeID	**string**	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: [https://kubernetes.io/docs/conc epts/storage/volumes#awselastic blockstore](https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore)

10.1.79. .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**	cachingMode is the Host Caching mode: None, Read Only, Read Write.
diskName	**string**	diskName is the Name of the data disk in the blob storage

Property	Type	Description
diskURI	**string**	diskURI is the URI of data disk in the blob storage
fsType	**string**	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
kind	**string**	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared
readOnly	**boolean**	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

10.1.80. .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**	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretName	**string**	secretName is the name of secret that contains Azure Storage Account Name and Key

Property	Type	Description
shareName	**string**	shareName is the azure share Name

10.1.81. .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 (string)**	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/ cephfs/README.md#how-to- use-it
path	**string**	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
readOnly	**boolean**	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/ cephfs/README.md#how-to- use-it
secretFile	**string**	secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/ cephfs/README.md#how-to- use-it

Property	Type	Description
secretRef	**object**	secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
user	**string**	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

10.1.82. .spec.config.volumes[].cephfs.secretRef

Description

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

Type

object

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

10.1.83. .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**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
readOnly	**boolean**	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	**object**	secretRef is optional: points to a secret object containing parameters used to connect to OpenStack.
volumeID	**string**	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md

10.1.84. .spec.config.volumes[].cinder.secretRef

Description

secretRef is optional: points to a secret object containing parameters used to connect to OpenStack.

Type

object

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

10.1.85. .spec.config.volumes[].configMap

Description

configMap represents a configMap that should populate this volume

Type

  object

Property	Type	Description
defaultMode	**integer**	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	**array**	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified 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 '..'.
items[]	**object**	Maps a string key to a path within a volume.
name	**string**	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
optional	**boolean**	optional specify whether the ConfigMap or its keys must be defined

10.1.86. .spec.config.volumes[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

10.1.87. .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**	key is the key to project.

Property	Type	Description
mode	**integer**	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

10.1.88. .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**	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.

Property	Type	Description
nodePublishSecretRef	**object**	nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.
readOnly	**boolean**	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
volumeAttributes	**object (string)**	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

10.1.89. .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. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

10.1.90. .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 file mode, like fsGroup, and the result can be other mode bits set.
items	**array**	Items is a list of downward API volume file
items[]	**object**	DownwardAPIVolumeFile represents information to create the file containing the pod field

10.1.91. .spec.config.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

array

10.1.92. .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, namespace and uid are supported.
mode	**integer**	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	**object**	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

10.1.93. .spec.config.volumes[].downwardAPI.items[].fieldRef

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

object

Required

- **fieldPath**

Property	Type	Description
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.

10.1.94. .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-or-string**	Specifies the output format of the exposed resources, defaults to "1"
resource	**string**	Required: resource to select

10.1.95. .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

Property	Type	Description
medium	**string**	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/conc epts/storage/volumes#emptydir
sizeLimit	**integer-or-string**	sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/conc epts/storage/volumes#emptydir

10.1.96. .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.

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.

10.1.97. .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	**object**	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.

10.1.98. .spec.config.volumes[].ephemeral.volumeClaimTemplate.metadata

Description

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.

Type

object

10.1.99. .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 (string)**	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**	dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Property	Type	Description
dataSourceRef	**object**	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Property	Type	Description
resources	**object**	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: [https://kubernetes.io/docs/conc epts/storage/persistent- volumes#resources](https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources)
selector	**object**	selector is a label query over volumes to consider for binding.
storageClassName	**string**	storageClassName is the name of the StorageClass required by the claim. More info: [https://kubernetes.io/docs/conc epts/storage/persistent- volumes#class-1](https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1)

Property	Type	Description
volumeAttributesClassName	**string**	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that no VolumeAttributesClass will be applied to the claim but it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be set by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).
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.

10.1.100. .spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

- **kind**

- **name**

Property	Type	Description
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

10.1.101. .spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef

Description

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

- **kind**

- **name**

Property	Type	Description
apiGroup	**string**	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	**string**	Kind is the type of resource being referenced
name	**string**	Name is the name of resource being referenced
namespace	**string**	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

10.1.102. .spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

Type

object

Property	Type	Description
limits	**integer-or-string**	Limits describes the maximum amount of compute resources allowed. More info: [https://kubernetes.io/docs/conc epts/configuration/manage-resources-containers/](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
requests	**integer-or-string**	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: [https://kubernetes.io/docs/conc epts/configuration/manage-resources-containers/](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

10.1.103. .spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector

Description

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

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.104. .spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector.match

Description

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

Type

array

10.1.105. .spec.config.volumes[].ephemeral.volumeClaimTemplate.spec.selector.match

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 (string)**	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.

10.1.106. .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**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
lun	**integer**	lun is Optional: FC target lun number
readOnly	**boolean**	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
targetWWNs	**array (string)**	targetWWNs is Optional: FC target worldwide names (WWNs)
wwids	**array (string)**	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

10.1.107. .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.

Property	Type	Description
fsType	**string**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
options	**object (string)**	options is Optional: this field holds extra command options if any.
readOnly	**boolean**	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	**object**	secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

10.1.108. .spec.config.volumes[].flexVolume.secretRef

Description

secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Type

object

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

10.1.109. .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**	datasetName is Name of the dataset stored as metadata → name on the dataset for Flocker should be considered as deprecated
datasetUUID	**string**	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

10.1.110. .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**	fsType is 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#gcepersistentdisk

Property	Type	Description
partition	**integer**	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
pdName	**string**	pdName is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	**boolean**	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

10.1.111. .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

Property	Type	Description
directory	**string**	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
repository	**string**	repository is the URL
revision	**string**	revision is the commit hash for the specified revision.

10.1.112. .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**	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	**string**	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

Property	Type	Description
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

10.1.113. .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

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

10.1.114. .spec.config.volumes[].image

Description

image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided:

- Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails.

- Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present.

- IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails.

The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (spec.containers[**].volumeMounts.mountPath**) **by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (spec.containers[**]**.volumeMounts.subpath). The field spec.securityContext.fsGroupChangePolicy has no effect on this volume type.

Type

object

Property	Type	Description
pullPolicy	**string**	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.

Property	Type	Description
reference	**string**	Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

10.1.115. .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**	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
chapAuthSession	**boolean**	chapAuthSession defines whether support iSCSI Session CHAP authentication

Property	Type	Description
fsType	**string**	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	**string**	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface <target portal>:<volume name> will be created for the connection.
iqn	**string**	iqn is the target iSCSI Qualified Name.
iscsiInterface	**string**	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
lun	**integer**	lun represents iSCSI Target Lun number.
portals	**array (string)**	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
readOnly	**boolean**	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
secretRef	**object**	secretRef is the CHAP Secret for iSCSI target and initiator authentication
targetPortal	**string**	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

10.1.116. .spec.config.volumes[].iscsi.secretRef

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

Type

object

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

10.1.117. .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

10.1.118. .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**	readOnly Will force the ReadOnly setting in VolumeMounts. Default false.

10.1.119. .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**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
pdID	**string**	pdID is the ID that identifies Photon Controller persistent disk

10.1.120. .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. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	**boolean**	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	**string**	volumeID uniquely identifies a Portworx volume

10.1.121. .spec.config.volumes[].projected

Description

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

Type

object

Property	Type	Description
defaultMode	**integer**	defaultMode are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
sources	**array**	sources is the list of volume projections. Each entry in this list handles one source.
sources[]	**object**	Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

10.1.122. .spec.config.volumes[].projected.sources

Description

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

Type

array

10.1.123. .spec.config.volumes[].projected.sources[]

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

object

Property	Type	Description

Property	Type	Description
clusterTrustBundle	**object**	ClusterTrustBundle allows a pod to access the **.spec.trustBundle** field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.
configMap	**object**	configMap information about the configMap data to project
downwardAPI	**object**	downwardAPI information about the downwardAPI data to project
secret	**object**	secret information about the secret data to project
serviceAccountToken	**object**	serviceAccountToken is information about the serviceAccountToken data to project

10.1.124. .spec.config.volumes[].projected.sources[].clusterTrustBundle

Description

ClusterTrustBundle allows a pod to access the **.spec.trustBundle** field of ClusterTrustBundle objects in an auto-updating file.
Alpha, gated by the ClusterTrustBundleProjection feature gate.

ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector.

Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem.

Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.

Type

object

Required

- **path**

Property	Type	Description
labelSelector	**object**	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".
name	**string**	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
optional	**boolean**	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
path	**string**	Relative path from the volume root to write the bundle.
signerName	**string**	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

10.1.125. .spec.config.volumes[].projected.sources[].clusterTrustBundle.labelSelector

Description

Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Type

object

Property	Type	Description
matchExpressions	**array**	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchExpressions[]	**object**	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	**object (string)**	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.

10.1.126. .spec.config.volumes[].projected.sources[].clusterTrustBundle.labelSelector.r

Description

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

Type

array

10.1.127. .spec.config.volumes[].projected.sources[].clusterTrustBundle.labelSelector.r

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 (string)**	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.

10.1.128. .spec.config.volumes[].projected.sources[].configMap

Description

configMap information about the configMap data to project

Type

object

Property	Type	Description
items	**array**	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	**object**	Maps a string key to a path within a volume.

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

10.1.129. .spec.config.volumes[].projected.sources[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

10.1.130. .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**	key is the key to project.

Property	Type	Description
mode	**integer**	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

10.1.131. .spec.config.volumes[].projected.sources[].downwardAPI

Description

downwardAPI information about the downwardAPI data to project

Type

object

Property	Type	Description
items	**array**	Items is a list of DownwardAPIVolume file
items[]	**object**	DownwardAPIVolumeFile represents information to create the file containing the pod field

10.1.132. .spec.config.volumes[].projected.sources[].downwardAPI.items

Description

Items is a list of DownwardAPIVolume file

Type

array

10.1.133. .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, namespace and uid are supported.
mode	**integer**	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	**object**	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

10.1.134. .spec.config.volumes[].projected.sources[].downwardAPI.items[].fieldRef

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

object

Required

- **fieldPath**

Property	Type	Description
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.

10.1.135. .spec.config.volumes[].projected.sources[].downwardAPI.items[].resourceFiel

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-or-string**	Specifies the output format of the exposed resources, defaults to "1"
resource	**string**	Required: resource to select

10.1.136. .spec.config.volumes[].projected.sources[].secret

Description

secret information about the secret data to project

Type

object

Property	Type	Description
items	**array**	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified 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 '..'.
items[]	**object**	Maps a string key to a path within a volume.
name	**string**	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	**boolean**	optional field specify whether the Secret or its key must be defined

10.1.137. .spec.config.volumes[].projected.sources[].secret.items

Description

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

10.1.138. .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**	key is the key to project.
mode	**integer**	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

10.1.139. .spec.config.volumes[].projected.sources[].serviceAccountToken

Description

serviceAccountToken is information about the serviceAccountToken data to project

Type

object

Required

- **path**

Property	Type	Description

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.

10.1.140. .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

Property	Type	Description
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.

10.1.141. .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

Property	Type	Description
fsType	**string**	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/concepts/storage/volumes#rbd
image	**string**	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	**string**	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitors	**array (string)**	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	**string**	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	**boolean**	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	**object**	secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Property	Type	Description
user	**string**	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

10.1.142. .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. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

10.1.143. .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

Property	Type	Description
fsType	**string**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
gateway	**string**	gateway is the host address of the ScaleIO API Gateway.
protectionDomain	**string**	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
readOnly	**boolean**	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	**object**	secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.
sslEnabled	**boolean**	sslEnabled Flag enable/disable SSL communication with Gateway, default false
storageMode	**string**	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
storagePool	**string**	storagePool is the ScaleIO Storage Pool associated with the protection domain.
system	**string**	system is the name of the storage system as configured in ScaleIO.
volumeName	**string**	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

10.1.144. .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. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: [https://kubernetes.io/docs/conc epts/overview/working-with- objects/names/#names](https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names)

10.1.145. .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**	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
items	**array**	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	**object**	Maps a string key to a path within a volume.
optional	**boolean**	optional field specify whether the Secret or its keys must be defined
secretName	**string**	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

10.1.146. .spec.config.volumes[].secret.items

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

10.1.147. .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**	key is the key to project.
mode	**integer**	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	**string**	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

10.1.148. .spec.config.volumes[].storageos

Description

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

Type

object

Property	Type	Description
fsType	**string**	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	**boolean**	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Property	Type	Description
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.

10.1.149. .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. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

10.1.150. .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**	fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
storagePolicyID	**string**	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
storagePolicyName	**string**	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
volumePath	**string**	volumePath is the path that identifies vSphere volume vmdk

10.1.151. .status

Description

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.

Property	Type	Description
catalogHealth[]	**object**	SubscriptionCatalogHealth describes the health of a CatalogSource the Subscription knows about.
conditions	**array**	Conditions is a list of the latest available observations about a Subscription's current state.
conditions[]	**object**	SubscriptionCondition represents the latest available observations of a Subscription's state.
currentCSV	**string**	CurrentCSV is the CSV the Subscription is progressing to.
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

10.1.152. .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

10.1.153. .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

10.1.154. .status.catalogHealth[].catalogSourceRef

Description

CatalogSourceRef is a reference to a CatalogSource.

Type

object

Property	Type	Description
apiVersion	**string**	API version of the referent.

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.
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: 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

10.1.155. .status.conditions

Description

Conditions is a list of the latest available observations about a Subscription's current state.

Type

array

10.1.156. .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.

10.1.157. .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.
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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency

Property	Type	Description
uid	**string**	UID of the referent. More info: [https://kubernetes.io/docs/conc epts/overview/working-with-objects/names/#uids](https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids)

10.1.158. .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**	
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.

10.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/operators.coreos.com/v1alpha1/subscriptions**

    - **GET**: list objects of kind Subscription

- **/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions**

- **DELETE**: delete collection of Subscription

- **GET**: list objects of kind Subscription

- **POST**: create a Subscription

- **/apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions/{name}**

  - **DELETE**: delete a 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

10.2.1. /apis/operators.coreos.com/v1alpha1/subscriptions

HTTP method

    GET

Description

    list objects of kind Subscription

Table 10.1. HTTP responses

HTTP code	Reponse body
200 – OK	**SubscriptionList** schema
401 – Unauthorized	Empty

10.2.2. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions

HTTP method

    DELETE

Description

    delete collection of Subscription

Table 10.2. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema

HTTP code	Reponse body
401 – Unauthorized	Empty

HTTP method

GET

Description

list objects of kind Subscription

Table 10.3. HTTP responses

HTTP code	Reponse body
200 – OK	**SubscriptionList** schema
401 – Unauthorized	Empty

HTTP method

POST

Description

create a Subscription

Table 10.4. 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.

Table 10.5. Body parameters

Parameter	Type	Description
body	**Subscription** schema	

Table 10.6. HTTP responses

HTTP code	Reponse body
200 – OK	**Subscription** schema
201 – Created	**Subscription** schema
202 – Accepted	**Subscription** schema
401 – Unauthorized	Empty

10.2.3. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions

Table 10.7. Global path parameters

Parameter	Type	Description
name	**string**	name of the Subscription

HTTP method

DELETE

Description

delete a Subscription

Table 10.8. 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

Table 10.9. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

read the specified Subscription

Table 10.10. HTTP responses

HTTP code	Reponse body
200 – OK	**Subscription** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Subscription

Table 10.11. 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.

Table 10.12. HTTP responses

HTTP code	Reponse body
200 - OK	**Subscription** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Subscription

Table 10.13. Query parameters

Parameter	Type	Description

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.

Table 10.14. Body parameters

Parameter	Type	Description
body	**Subscription** schema	

Table 10.15. HTTP responses

HTTP code	Reponse body
200 – OK	**Subscription** schema
201 – Created	**Subscription** schema
401 – Unauthorized	Empty

10.2.4. /apis/operators.coreos.com/v1alpha1/namespaces/{namespace}/subscriptions

Table 10.16. Global path parameters

Parameter	Type	Description
name	**string**	name of the Subscription

HTTP method

GET

Description

read status of the specified Subscription

Table 10.17. HTTP responses

HTTP code	Reponse body
200 – OK	**Subscription** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Subscription

Table 10.18. 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.

Table 10.19. HTTP responses

HTTP code	Reponse body
200 – OK	**Subscription** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Subscription

Table 10.20. 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.

Table 10.21. Body parameters

Parameter	Type	Description
body	**Subscription** schema	

Table 10.22. HTTP responses

HTTP code	Reponse body
200 – OK	**Subscription** schema
201 – Created	**Subscription** schema
401 – Unauthorized	Empty