

Workload APIs

[DaemonSet \[apps/v1\]](#)[Deployments \[apps/v1\]](#)

DaemonSet [apps/v1]

Description

DaemonSet represents the configuration of a daemon set.

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ^
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ^
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	DaemonSetSpec is the specification of a daemon set.
<code>status</code>	<code>object</code>	DaemonSetStatus represents the current status of a daemon set.

.spec

Description

DaemonSetSpec is the specification of a daemon set.

Type

object

Required

selector

template

Property	Type	Description
<code>minReadySeconds</code>	<code>integer</code>	The minimum number of seconds for which a newly created DaemonSet 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).
<code>revisionHistoryLimit</code>	<code>integer</code>	The number of old history to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 10.
<code>selector</code>	<code>object</code>	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.
<code>template</code>	<code>object</code>	PodTemplateSpec describes the data a pod should have when created from a template
<code>updateStrategy</code>	<code>object</code>	DaemonSetUpdateStrategy is a struct used to control the update strategy for a DaemonSet.

`.spec.selector`

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

`.spec.selector.matchExpressions`

Description

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

Type

`array`

`.spec.selector.matchExpressions[]`

Description

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

Type

object

Required

key operator

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

.spec.selector.matchExpressions[].values

Description

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

Type

array

.spec.selector.matchExpressions[].values[]

Type

string

.spec.selector.matchLabels

Description

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

Type

object

.spec.template

Description

PodTemplateSpec describes the data a pod should have when created from a template

Type

object

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	PodSpec is a description of a pod.

.spec.template.spec

Description

PodSpec is a description of a pod.

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	Affinity is a group of affinity scheduling rules.
automountServiceAccountToken	boolean	AutomountServiceAccountToken indicates whether a service account token should be automatically mounted.
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.
dnsConfig	object	PodDNSConfig defines the DNS parameters of a pod in addition to those generated from 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'. Possible enum values: "ClusterFirst" indicates that the pod should use cluster DNS first unless hostNetwork is true, if it is available, then fall back on the default (as determined by kubelet) DNS settings. "ClusterFirstWithHostNet" indicates that the pod should use cluster DNS first, if it is available, then fall back on the default (as determined by kubelet) DNS settings.

Property	Type	Description
		"Default" indicates that the pod should use the default (as determined by kubelet) DNS settings. "None" indicates that the pod should use empty DNS settings. DNS parameters such as nameservers and search paths should be defined via DNSConfig.
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.
hostAliases	array	HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
hostIPC	boolean	Use the host's ipc namespace. Optional: Default to false.
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 users 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/concepts/containers/images#specifying-imagepullsecrets-on-a-pod

Property	Type	Description
<code>initContainers</code>	<code>array</code>	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/ ↗
<code>nodeName</code>	<code>string</code>	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/docs/concepts/scheduling-eviction/assign-pod-node/#nodename ↗
<code>nodeSelector</code>	<code>object</code>	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/ ↗
<code>os</code>	<code>object</code>	PodOS defines the OS parameters of a pod.
<code>overhead</code>	<code>object</code>	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 ↗
<code>preemptionPolicy</code>	<code>string</code>	PreemptionPolicy is the Policy for preempting pods with lower priority. One of Never, PreemptLowerPriority. Defaults to PreemptLowerPriority if unset. Possible enum values: <code>"Never"</code> means that pod never preempts other pods with lower priority. <code>"PreemptLowerPriority"</code> means that pod can preempt other pods with lower priority.
<code>priority</code>	<code>integer</code>	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
<code>priorityClassName</code>	<code>string</code>	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.
<code>readinessGates</code>	<code>array</code>	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 ↗
<code>resourceClaims</code>	<code>array</code>	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.
<code>resources</code>	<code>object</code>	ResourceRequirements describes the compute resource requirements.
<code>restartPolicy</code>	<code>string</code>	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/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy ↗ Possible enum values: <code>"Always"</code> <code>"Never"</code> <code>"OnFailure"</code>
<code>runtimeClassName</code>	<code>string</code>	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 ↗
<code>schedulerName</code>	<code>string</code>	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
<code>schedulingGates</code>	<code>array</code>	SchedulingGates is an opaque list of values that 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.

Property	Type	Description
<code>securityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
<code>serviceAccount</code>	<code>string</code>	DeprecatedServiceAccount is a deprecated alias for ServiceAccountName. Deprecated: Use serviceAccountName instead.
<code>serviceAccountName</code>	<code>string</code>	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
<code>setHostnameAsFQDN</code>	<code>boolean</code>	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.
<code>shareProcessNamespace</code>	<code>boolean</code>	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.
<code>subdomain</code>	<code>string</code>	If specified, the fully qualified Pod hostname will be "...svc.". If not specified, the pod will not have a domainname at all.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no 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.
<code>tolerations</code>	<code>array</code>	If specified, the pod's tolerations.
<code>topologySpreadConstraints</code>	<code>array</code>	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.

Property	Type	Description
<code>volumes</code>	<code>array</code>	List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes ^

`.spec.template.spec.affinity`

Description

Affinity is a group of affinity scheduling rules.

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Node affinity is a group of node affinity scheduling rules.
<code>podAffinity</code>	<code>object</code>	Pod affinity is a group of inter pod affinity scheduling rules.
<code>podAntiAffinity</code>	<code>object</code>	Pod anti affinity is a group of inter pod anti affinity scheduling rules.

`.spec.template.spec.affinity.nodeAffinity`

Description

Node affinity is a group of node affinity scheduling rules.

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, 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.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

`.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

`array`

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

Description

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

Type

`object`

Required

`weight``preference`

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

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

Description

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

Type

`object`

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

Type

object

Required

nodeSelectorTerms

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

.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

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

Type

array

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

Description

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

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

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

Description

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

Type

array

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

Type

string

.spec.template.spec.affinity.podAffinity

Description

Pod affinity is a group of inter pod affinity scheduling rules.

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

Property	Type	Description
		corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
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.

.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

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

Description

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

Type

object

Required

weight podAffinityTerm

Property	Type	Description
podAffinityTerm	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 matches that of any node on which a pod of the set of pods is running
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

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

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 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.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key not 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 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 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.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

keyoperator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Type

string

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key

operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

`.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

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

Description

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

Type

object

Required

topologyKey

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

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

.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution.labelSelector

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution.labelSelector.matchExpressions

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

`array`

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

Type

`string`

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

Description

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

Type

`array`

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

Type

`string`

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

`object`

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

`.spec.template.spec.affinity.podAntiAffinity`

Description

Pod anti affinity is a group of inter pod anti affinity scheduling rules.

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	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

Property	Type	Description
		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.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	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.

.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

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

Description

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

Type

`object`

Required

`weight` `podAffinityTerm`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	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 matches that of any node on which a pod of the set of pods is running
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

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

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 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.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
namespaceSelector	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.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

keyoperator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Type

string

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

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

Description

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

Type

object

Required

topologyKey

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

Property	Type	Description
		consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label 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.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

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

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

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

Description

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

Type

`array`

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Type

string

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

.spec.template.spec.containers

Description

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

Type

array

.spec.template.spec.containers[]

Description

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

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must 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.
image	string	Container image name. 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.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: "Always" means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present

Property	Type	Description
<code>lifecycle</code>	<code>object</code>	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
<code>livenessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>name</code>	<code>string</code>	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
<code>ports</code>	<code>array</code>	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. 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.
<code>readinessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	ResourceRequirements describes the compute resource requirements.
<code>restartPolicy</code>	<code>string</code>	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers 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.
<code>securityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>startupProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
<code>stdin</code>	<code>boolean</code>	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.
<code>stdinOnce</code>	<code>boolean</code>	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
<code>terminationMessagePath</code>	<code>string</code>	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.
<code>terminationMessagePolicy</code>	<code>string</code>	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. Possible enum values: <code>"FallbackToLogsOnError"</code> will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. <code>"File"</code> is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used by the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

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

Description

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

Type

array

.spec.template.spec.containers[].args[]**Type**

string

.spec.template.spec.containers[].command**Description**

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

Type

array

.spec.template.spec.containers[].command[]**Type**

string

.spec.template.spec.containers[].env**Description**

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

Type

array

.spec.template.spec.containers[].env[]**Description**

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

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

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

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

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

Description

Selects a key from a ConfigMap.

Type

object

Required

key

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

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

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

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

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

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number less than 10⁻³⁰. When a Quantity is parsed from a string, it will remember the type of suffix it had, and will use that type when serializing. When serializing, Quantity will be put in "canonical form". This means that Exponent is used for all values greater than or equal to 10³. - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be emitted for all values greater than or equal to 10³. The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. All internal calculations are done on integers. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form when serialized. This format is intended to make it difficult to use these numbers without writing software that understands the format.
resource	string	Required: resource to select

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

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.

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

.spec.template.spec.containers[].envFrom

Description

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

Type

array

.spec.template.spec.containers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

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

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

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

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

Description

SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

`object`

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

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

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
<code>preStop</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

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

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
<code>exec</code>	object	ExecAction describes a "run in container" action.
<code>httpGet</code>	object	HTTPGetAction describes an action based on HTTP Get requests.
<code>sleep</code>	object	SleepAction describes a "sleep" action.
<code>tcpSocket</code>	object	TCPSocketAction describes an action based on opening a socket

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

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

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

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

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

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

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

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.containers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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	TCPSocketAction describes an action based on opening a socket
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

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

Property	Type	Description
value	string	The header field value

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.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

.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.

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

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

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	ExecAction describes a "run in container" action.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPCAction specifies an action involving a GRPC service.
<code>httpGet</code>	<code>object</code>	HTTPGetAction describes an action based on HTTP Get requests.
<code>initialDelaySeconds</code>	<code>integer</code>	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
<code>periodSeconds</code>	<code>integer</code>	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	TCPSocketAction describes an action based on opening a socket
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

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

Property	Type	Description
		Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

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

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.containers[].resizePolicy

Description

Resources resize policy for the container.

Type

`array`

.spec.template.spec.containers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

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

.spec.template.spec.containers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
<code>claims</code>	<code>array</code>	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

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

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

Description

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

Type

`array`

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

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

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

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

Description

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

Type

`object`

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

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.containers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	object	AppArmorProfile defines a pod or container's AppArmor settings.
<code>capabilities</code>	object	Adds and removes POSIX capabilities from running containers.
<code>privileged</code>	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 <code>spec.os.name</code> is windows.
<code>procMount</code>	string	<code>procMount</code> denotes the type of proc mount to use for the containers. The default value is <code>Default</code> which uses the container runtime defaults for readonly paths and masked paths. This requires the <code>ProcMountType</code> feature flag to be enabled. Note that this field cannot be set when <code>spec.os.name</code> is windows. Possible enum values: <code>"Default"</code> uses the container runtime defaults for readonly and masked paths for <code>/proc</code>. Most container runtimes mask certain paths in <code>/proc</code> to avoid accidental security exposure of special devices or information. <code>"Unmasked"</code> bypasses the default masking behavior of the container runtime and ensures the newly created <code>/proc</code> the container stays in tact with no modifications.
<code>readOnlyRootFilesystem</code>	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	SELinuxOptions are the labels to be applied to the container
<code>seccompProfile</code>	<code>object</code>	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
<code>windowsOptions</code>	<code>object</code>	WindowsSecurityContextOptions contain Windows-specific options and credentials.

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

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

`object`

Required

`type`

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

Property	Type	Description
		"<code>RuntimeDefault</code>" indicates that the container runtime's default AppArmor profile should be used. "<code>Unconfined</code>" indicates that no AppArmor profile should be enforced.

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

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array	Added capabilities
drop	array	Removed capabilities

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

Description

Added capabilities

Type

array

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

Type

string

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

Description

Removed capabilities

Type

array

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

Type

string

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

Description

SELinuxOptions are the labels to be applied to the container

Type

object

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

.spec.template.spec.containers[].securityContext.seccompProfile**Description**

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	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.
<code>type</code>	<code>string</code>	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. Possible enum values: <code>"Localhost"</code> indicates a profile defined in a file on the node should be used. The file's location relative to /seccomp. <code>"RuntimeDefault"</code> represents the default container runtime seccomp profile. <code>"Unconfined"</code> indicates no seccomp profile is applied (A.K.A. unconfined).

.spec.template.spec.containers[].securityContext.windowsOptions**Description**

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

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

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	ExecAction describes a "run in container" action.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPCAction specifies an action involving a GRPC service.
<code>httpGet</code>	<code>object</code>	HTTPGetAction describes an action based on HTTP Get requests.
<code>initialDelaySeconds</code>	<code>integer</code>	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 ↗

Property	Type	Description
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	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.
<code>tcpSocket</code>	<code>object</code>	TCPSocketAction describes an action based on opening a socket
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	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 <code>terminationGracePeriodSeconds</code> 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 <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used if unset.
<code>timeoutSeconds</code>	<code>integer</code>	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

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

Description

ExecAction describes a "run in container" action.

Type

`object`

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

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

Description

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

Type

`array`

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

Type

string

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

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

Property	Type	Description
		Scheme to use for connecting to the host. Defaults to HTTP.
scheme	string	Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

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

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

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

Description

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

Type

`array`

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

Description

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

Type

`object`

Required

`name` `devicePath`

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

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

Description

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

Type

`array`

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

Description

`VolumeMount` describes a mounting of a `Volume` within a container.

Type

`object`

Required

`name` `mountPath`

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain '.'.
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>). Possible enum values: <code>"Bidirectional"</code> means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). <code>"HostToContainer"</code> means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). <code>"None"</code> means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be unspecified, which defaults to <code>None</code>). If this field is not specified, it is treated as an equivalent of <code>Disabled</code>.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to <code>SubPath</code> but environment variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. Defaults to "" (volume's root). <code>SubPathExpr</code> and <code>SubPath</code> are mutually exclusive.

Description

PodDNSConfig defines the DNS parameters of a pod in addition to those generated from DNSPolicy.

Type

object

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

.spec.template.spec.dnsConfig.nameservers

Description

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

Type

array

.spec.template.spec.dnsConfig.nameservers[]

Type

string

.spec.template.spec.dnsConfig.options

Description

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

Type

array

.spec.template.spec.dnsConfig.options[]

Description

PodDNSConfigOption defines DNS resolver options of a pod.

Type

object

Property	Type	Description
name	string	Name is this DNS resolver option's name. Required.
value	string	Value is this DNS resolver option's value.

.spec.template.spec.dnsConfig.searches

Description

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

Type

array

.spec.template.spec.dnsConfig.searches[]

Type

string

.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

.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	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

Property	Type	Description
		exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell ↗
command	array	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(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.
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.
image	string	Container image name. More info: 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/concepts/containers/images#updating-images ↗ Possible enum values: "Always" means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
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.

Property	Type	Description
<code>ports</code>	<code>array</code>	Ports are not allowed for ephemeral containers.
<code>readinessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	ResourceRequirements describes the compute resource requirements.
<code>restartPolicy</code>	<code>string</code>	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.
<code>securityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>startupProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>stdin</code>	<code>boolean</code>	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.
<code>stdinOnce</code>	<code>boolean</code>	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
<code>targetContainerName</code>	<code>string</code>	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.
<code>terminationMessagePath</code>	<code>string</code>	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

Property	Type	Description
		length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.
		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.
terminationMessagePolicy	string	Possible enum values: "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
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.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.
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.

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

Description

Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info:

<https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

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

Type

string

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

Description

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

Type

array

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

Type

string

.spec.template.spec.ephemeralContainers[].env

Description

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

Type

array

.spec.template.spec.ephemeralContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

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

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

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

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

Description

Selects a key from a ConfigMap.

Type

object

Required

key

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

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

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

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

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

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number. When a Quantity is parsed from a string, it will remember the type of suffix it had, and the precision of the number. Before serializing, Quantity will be put in "canonical form". This means that Exponent notation will be used when the number is not an integer, the exponent is not zero, and the number is not too small to fit in the normal decimal representation. - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be emitted when the number is not an integer, the exponent is not zero, and the number is not too small to fit in the normal decimal representation. The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. It is always an integer of the appropriate unit. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form when serialized. This format is intended to make it difficult to use these numbers without writing something like:
resource	string	Required: resource to select

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

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.

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

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

Description

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

Type

`array`

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

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

`object`

Property	Type	Description
<code>configMapRef</code>	<code>object</code>	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
<code>prefix</code>	<code>string</code>	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	<code>object</code>	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

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

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

`object`

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

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

Description

SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

`object`

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

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

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
<code>preStop</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

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

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
<code>exec</code>	object	ExecAction describes a "run in container" action.
<code>httpGet</code>	object	HTTPGetAction describes an action based on HTTP Get requests.
<code>sleep</code>	object	SleepAction describes a "sleep" action.
<code>tcpSocket</code>	object	TCPSocketAction describes an action based on opening a socket

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

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

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

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

Description

SleepAction describes a "sleep" action.

Type

`object`

Required

`seconds`

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

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

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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	TCPSocketAction describes an action based on opening a socket
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

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

Property	Type	Description
value	string	The header field value

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.ephemeralContainers[].ports

Description

Ports are not allowed for ephemeral containers.

Type

array

.spec.template.spec.ephemeralContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

containerPort

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

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

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

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	ExecAction describes a "run in container" action.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPCAction specifies an action involving a GRPC service.
<code>httpGet</code>	<code>object</code>	HTTPGetAction describes an action based on HTTP Get requests.
<code>initialDelaySeconds</code>	<code>integer</code>	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
<code>periodSeconds</code>	<code>integer</code>	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	TCPSocketAction describes an action based on opening a socket
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

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

Property	Type	Description
		Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

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

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.ephemeralContainers[].resizePolicy

Description

Resources resize policy for the container.

Type

`array`

.spec.template.spec.ephemeralContainers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

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

.spec.template.spec.ephemeralContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
<code>claims</code>	<code>array</code>	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

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

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

Description

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

Type

`array`

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

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

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

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

Description

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

Type

`object`

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

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.ephemeralContainers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	object	AppArmorProfile defines a pod or container's AppArmor settings.
<code>capabilities</code>	object	Adds and removes POSIX capabilities from running containers.
<code>privileged</code>	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 <code>spec.os.name</code> is windows.
<code>procMount</code>	string	<code>procMount</code> denotes the type of proc mount to use for the containers. The default value is <code>Default</code> which uses the container runtime defaults for readonly paths and masked paths. This requires the <code>ProcMountType</code> feature flag to be enabled. Note that this field cannot be set when <code>spec.os.name</code> is windows. Possible enum values: <code>"Default"</code> uses the container runtime defaults for readonly and masked paths for <code>/proc</code>. Most container runtimes mask certain paths in <code>/proc</code> to avoid accidental security exposure of special devices or information. <code>"Unmasked"</code> bypasses the default masking behavior of the container runtime and ensures the newly created <code>/proc</code> the container stays in tact with no modifications.
<code>readOnlyRootFilesystem</code>	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

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

Description

AppArmorProfile defines a pod or container's AppArmor settings.

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. Possible enum values: "Localhost" indicates that a profile pre-loaded on the node should be used.

Property	Type	Description
		"<code>RuntimeDefault</code>" indicates that the container runtime's default AppArmor profile should be used. "<code>Unconfined</code>" indicates that no AppArmor profile should be enforced.

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

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array	Added capabilities
drop	array	Removed capabilities

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

Description

Added capabilities

Type

array

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

Type

string

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

Description

Removed capabilities

Type

array

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

Type

string

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

Description

SELinuxOptions are the labels to be applied to the container

Type

object

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

.spec.template.spec.ephemeralContainers[].securityContext.seccompProfile**Description**

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	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.
<code>type</code>	<code>string</code>	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. Possible enum values: <code>"Localhost"</code> indicates a profile defined in a file on the node should be used. The file's location relative to /seccomp. <code>"RuntimeDefault"</code> represents the default container runtime seccomp profile. <code>"Unconfined"</code> indicates no seccomp profile is applied (A.K.A. unconfined).

.spec.template.spec.ephemeralContainers[].securityContext.windowsOptions**Description**

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.template.spec.ephemeralContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	ExecAction describes a "run in container" action.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPCAction specifies an action involving a GRPC service.
<code>httpGet</code>	<code>object</code>	HTTPGetAction describes an action based on HTTP Get requests.
<code>initialDelaySeconds</code>	<code>integer</code>	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 ↗

Property	Type	Description
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	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.
<code>tcpSocket</code>	<code>object</code>	TCPSocketAction describes an action based on opening a socket
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	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 <code>terminationGracePeriodSeconds</code> 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 <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used if unset.
<code>timeoutSeconds</code>	<code>integer</code>	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

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

Description

ExecAction describes a "run in container" action.

Type

`object`

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

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

Description

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

Type

`array`

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

Type

string

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

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

Property	Type	Description
		Scheme to use for connecting to the host. Defaults to HTTP.
scheme	string	Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

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

Property	Type	Description
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.ephemeralContainers[].volumeDevices

Description

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

Type

array

.spec.template.spec.ephemeralContainers[].volumeDevices[]

Description

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

Type

object

Required

name devicePath

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

.spec.template.spec.ephemeralContainers[].volumeMounts

Description

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

Type

array

.spec.template.spec.ephemeralContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

name mountPath

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain '.'.
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>). Possible enum values: <code>"Bidirectional"</code> means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). <code>"HostToContainer"</code> means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). <code>"None"</code> means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be unspecified, which defaults to <code>None</code>). If this field is not specified, it is treated as an equivalent of <code>Disabled</code>.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to <code>SubPath</code> but environment variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. Defaults to "" (volume's root). <code>SubPathExpr</code> and <code>SubPath</code> are mutually exclusive.

Description

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

Type

array

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

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

Type

object

Required

ip

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

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

Hostnames for the above IP address.

Type

array

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

string

.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

.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 ^

.spec.template.spec.initContainers

Description

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

Type

array

.spec.template.spec.initContainers[]

Description

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

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" 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
<code>command</code>	<code>array</code>	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
<code>env</code>	<code>array</code>	List of environment variables to set in the container. Cannot be updated.
<code>envFrom</code>	<code>array</code>	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.
<code>image</code>	<code>string</code>	Container image name. 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.
<code>imagePullPolicy</code>	<code>string</code>	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: <code>"Always"</code> means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. <code>"IfNotPresent"</code> means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. <code>"Never"</code> means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
<code>lifecycle</code>	<code>object</code>	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
<code>livenessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>name</code>	<code>string</code>	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.

Property	Type	Description
<code>ports</code>	<code>array</code>	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. 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.
<code>readinessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	ResourceRequirements describes the compute resource requirements.
<code>restartPolicy</code>	<code>string</code>	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers 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.
<code>securityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>startupProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>stdin</code>	<code>boolean</code>	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.
<code>stdinOnce</code>	<code>boolean</code>	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
<code>terminationMessagePath</code>	<code>string</code>	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.
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be populated. File will use the contents of <code>terminationMessagePath</code> to populate the container status message on both success and failure. <code>FallbackToLogsOnError</code> 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. Possible enum values: <code>"FallbackToLogsOnError"</code> will read the most recent contents of the container logs for the container status message when the container exits with an error and the <code>terminationMessagePath</code> has no contents. <code>"File"</code> is the default behavior and will set the container status message to the contents of the container's <code>terminationMessagePath</code> when the container exits.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	<code>volumeDevices</code> is the list of block devices to be used by the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

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

Description

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

Type

`array`

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

Type

string

.spec.template.spec.initContainers[].command

Description

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

Type

array

.spec.template.spec.initContainers[].command[]

Type

string

.spec.template.spec.initContainers[].env

Description

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

Type

array

.spec.template.spec.initContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

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

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

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
<code>configMapKeyRef</code>	object	Selects a key from a ConfigMap.
<code>fieldRef</code>	object	ObjectFieldSelector selects an APIVersioned field of an object.
<code>resourceFieldRef</code>	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
<code>secretKeyRef</code>	object	SecretKeySelector selects a key of a Secret.

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

Description

Selects a key from a ConfigMap.

Type

object

Required

key

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

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

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

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

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

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number. When a Quantity is parsed from a string, it will remember the type of suffix it had, and will emit it when serializing. Before serializing, Quantity will be put in "canonical form". This means that Exponent notation will be used when appropriate. - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be emitted when appropriate The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. It is always an integer of the appropriate unit. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form when serialized. This format is intended to make it difficult to use these numbers without writing something like:
resource	string	Required: resource to select

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

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.

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

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

Description

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

Type

`array`

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

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

`object`

Property	Type	Description
<code>configMapRef</code>	<code>object</code>	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
<code>prefix</code>	<code>string</code>	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
<code>secretRef</code>	<code>object</code>	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

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

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

`object`

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

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

Description

SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

`object`

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

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

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
<code>preStop</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

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

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
<code>exec</code>	object	ExecAction describes a "run in container" action.
<code>httpGet</code>	object	HTTPGetAction describes an action based on HTTP Get requests.
<code>sleep</code>	object	SleepAction describes a "sleep" action.
<code>tcpSocket</code>	object	TCPSocketAction describes an action based on opening a socket

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

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

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

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

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

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

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	ExecAction describes a "run in container" action.
<code>httpGet</code>	<code>object</code>	HTTPGetAction describes an action based on HTTP Get requests.
<code>sleep</code>	<code>object</code>	SleepAction describes a "sleep" action.
<code>tcpSocket</code>	<code>object</code>	TCPSocketAction describes an action based on opening a socket

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

Description

ExecAction describes a "run in container" action.

Type

`object`

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

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

Description

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

Type

`array`

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

Type

`string`

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

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

Description

SleepAction describes a "sleep" action.

Type

`object`

Required

`seconds`

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

`.spec.template.spec.initContainers[].livenessProbe`

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
<code>exec</code>	object	ExecAction describes a "run in container" action.
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
<code>grpc</code>	object	GRPCAction specifies an action involving a GRPC service.
<code>httpGet</code>	object	HTTPGetAction describes an action based on HTTP Get requests.
<code>initialDelaySeconds</code>	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
<code>periodSeconds</code>	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	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.
<code>tcpSocket</code>	object	TCPSocketAction describes an action based on opening a socket
<code>terminationGracePeriodSeconds</code>	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.
<code>timeoutSeconds</code>	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

`.spec.template.spec.initContainers[].livenessProbe.exec`

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

.spec.template.spec.initContainers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

.spec.template.spec.initContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

Property	Type	Description
<code>value</code>	<code>string</code>	The header field value

`.spec.template.spec.initContainers[].livenessProbe.tcpSocket`

Description

TCPSocketAction describes an action based on opening a socket

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

`.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`

`.spec.template.spec.initContainers[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

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

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

`.spec.template.spec.initContainers[].readinessProbe`

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	ExecAction describes a "run in container" action.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPCAction specifies an action involving a GRPC service.
<code>httpGet</code>	<code>object</code>	HTTPGetAction describes an action based on HTTP Get requests.
<code>initialDelaySeconds</code>	<code>integer</code>	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
<code>periodSeconds</code>	<code>integer</code>	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	TCPSocketAction describes an action based on opening a socket
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

.spec.template.spec.initContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

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

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

Description

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

Type

array

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

Type

string

.spec.template.spec.initContainers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

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

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

Property	Type	Description
		Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

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

Description

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

Type

array

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

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

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

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

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.initContainers[].resizePolicy

Description

Resources resize policy for the container.

Type

`array`

.spec.template.spec.initContainers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

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

.spec.template.spec.initContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
<code>claims</code>	<code>array</code>	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

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

`.spec.template.spec.initContainers[].resources.claims`

Description

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

Type

`array`

`.spec.template.spec.initContainers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

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

`.spec.template.spec.initContainers[].resources.limits`

Description

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

Type

`object`

`.spec.template.spec.initContainers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.initContainers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	object	AppArmorProfile defines a pod or container's AppArmor settings.
<code>capabilities</code>	object	Adds and removes POSIX capabilities from running containers.
<code>privileged</code>	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 <code>spec.os.name</code> is windows.
<code>procMount</code>	string	<code>procMount</code> denotes the type of proc mount to use for the containers. The default value is <code>Default</code> which uses the container runtime defaults for readonly paths and masked paths. This requires the <code>ProcMountType</code> feature flag to be enabled. Note that this field cannot be set when <code>spec.os.name</code> is windows. Possible enum values: <code>"Default"</code> uses the container runtime defaults for readonly and masked paths for <code>/proc</code>. Most container runtimes mask certain paths in <code>/proc</code> to avoid accidental security exposure of special devices or information. <code>"Unmasked"</code> bypasses the default masking behavior of the container runtime and ensures the newly created <code>/proc</code> the container stays in tact with no modifications.
<code>readOnlyRootFilesystem</code>	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	SELinuxOptions are the labels to be applied to the container
<code>seccompProfile</code>	<code>object</code>	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
<code>windowsOptions</code>	<code>object</code>	WindowsSecurityContextOptions contain Windows-specific options and credentials.

`.spec.template.spec.initContainers[].securityContext.appArmorProfile`

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

`object`

Required

`type`

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

Property	Type	Description
		"<code>RuntimeDefault</code>" indicates that the container runtime's default AppArmor profile should be used. "<code>Unconfined</code>" indicates that no AppArmor profile should be enforced.

.spec.template.spec.initContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array	Added capabilities
drop	array	Removed capabilities

.spec.template.spec.initContainers[].securityContext.capabilities.add

Description

Added capabilities

Type

array

.spec.template.spec.initContainers[].securityContext.capabilities.add[]

Type

string

.spec.template.spec.initContainers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

array

.spec.template.spec.initContainers[].securityContext.capabilities.drop[]

Type

string

.spec.template.spec.initContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

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

.spec.template.spec.initContainers[].securityContext.seccompProfile**Description**

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	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.
<code>type</code>	<code>string</code>	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. Possible enum values: <code>"Localhost"</code> indicates a profile defined in a file on the node should be used. The file's location relative to /seccomp. <code>"RuntimeDefault"</code> represents the default container runtime seccomp profile. <code>"Unconfined"</code> indicates no seccomp profile is applied (A.K.A. unconfined).

.spec.template.spec.initContainers[].securityContext.windowsOptions**Description**

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.template.spec.initContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
<code>exec</code>	<code>object</code>	ExecAction describes a "run in container" action.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPCAction specifies an action involving a GRPC service.
<code>httpGet</code>	<code>object</code>	HTTPGetAction describes an action based on HTTP Get requests.
<code>initialDelaySeconds</code>	<code>integer</code>	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 ↗

Property	Type	Description
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	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.
<code>tcpSocket</code>	<code>object</code>	TCPSocketAction describes an action based on opening a socket
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	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.
<code>timeoutSeconds</code>	<code>integer</code>	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

`.spec.template.spec.initContainers[].startupProbe.exec`

Description

ExecAction describes a "run in container" action.

Type

`object`

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

`.spec.template.spec.initContainers[].startupProbe.exec.command`

Description

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

Type

`array`

.spec.template.spec.initContainers[].startupProbe.exec.command[]

Type

string

.spec.template.spec.initContainers[].startupProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

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

.spec.template.spec.initContainers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

Property	Type	Description
		Scheme to use for connecting to the host. Defaults to HTTP.
scheme	string	Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders

Description

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

Type

array

.spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

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

.spec.template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

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

Property	Type	Description
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.initContainers[].volumeDevices

Description

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

Type

array

.spec.template.spec.initContainers[].volumeDevices[]

Description

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

Type

object

Required

name devicePath

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

.spec.template.spec.initContainers[].volumeMounts

Description

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

Type

array

.spec.template.spec.initContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

name mountPath

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain '.'.
<code>mountPropagation</code>	<code>string</code>	<code>mountPropagation</code> determines how mounts are propagated from the host to container and the other way around. When not set, <code>MountPropagationNone</code> is used. This field is beta in 1.10. When <code>RecursiveReadOnly</code> is set to <code>IfPossible</code> or to <code>Enabled</code>, <code>MountPropagation</code> must be <code>None</code> or unspecified (which defaults to <code>None</code>). Possible enum values: <code>"Bidirectional"</code> means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). <code>"HostToContainer"</code> means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). <code>"None"</code> means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
<code>name</code>	<code>string</code>	This must match the Name of a Volume.
<code>readOnly</code>	<code>boolean</code>	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
<code>recursiveReadOnly</code>	<code>string</code>	<code>RecursiveReadOnly</code> specifies whether read-only mounts should be handled recursively. If <code>ReadOnly</code> is false, this field has no meaning and must be unspecified. If <code>ReadOnly</code> is true, and this field is set to <code>Disabled</code>, the mount is not made recursively read-only. If this field is set to <code>IfPossible</code>, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to <code>Enabled</code>, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to <code>IfPossible</code> or <code>Enabled</code>, <code>MountPropagation</code> must be set to <code>None</code> (or be unspecified, which defaults to <code>None</code>). If this field is not specified, it is treated as an equivalent of <code>Disabled</code>.
<code>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to <code>SubPath</code> but environment variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. Defaults to "" (volume's root). <code>SubPathExpr</code> and <code>SubPath</code> are mutually exclusive.

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: <https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

object

.spec.template.spec.os

Description

PodOS defines the OS parameters of a pod.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration ↗ Clients should expect to handle additional values and treat unrecognized values in this field as os: null

.spec.template.spec.overhead

Description

Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info: <https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md>

Type

object

.spec.template.spec.readinessGates

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type

array

.spec.template.spec.readinessGates[]

Description

PodReadinessGate contains the reference to a pod condition

Type

object

Required

conditionType

Property	Type	Description
conditionType	string	ConditionType refers to a condition in the pod's condition list with matching type.

.spec.template.spec.resourceClaims

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.

Type

array

.spec.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.
resourceClaimTemplateName	string	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in pod.status.resourceClaimStatuses. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

.spec.template.spec.resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
claims	array	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.template.spec.resources.claims

Description

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

Type

array

.spec.template.spec.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

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

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

`.spec.template.spec.resources.limits`

Description

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

Type

`object`

`.spec.template.spec.resources.requests`

Description

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

Type

`object`

`.spec.template.spec.schedulingGates`

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

`array`

`.spec.template.spec.schedulingGates[]`

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the scheduling gate. Each scheduling gate must have a unique name field.

`.spec.template.spec.securityContext`

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	AppArmorProfile defines a pod or container's AppArmor settings.
<code>fsGroup</code>	<code>integer</code>	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
<code>fsGroupChangePolicy</code>	<code>string</code>	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. Possible enum values: • <code>"Always"</code> indicates that volume's ownership and permissions should always be changed whenever volume is mounted inside a Pod. This the default behavior. • <code>"OnRootMismatch"</code> indicates that volume's ownership and permissions will be changed only when permission and ownership of root directory does not match with expected permissions on the volume. This can help shorten the time it takes to change ownership and permissions of a volume.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in 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.

Property	Type	Description
<code>seLinuxChangePolicy</code>	<code>string</code>	<code>seLinuxChangePolicy</code> defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting <code>spec.seLinuxMount: true</code> in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same <code>seLinuxChangePolicy</code>, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>seLinuxOptions</code>	<code>object</code>	SELinuxOptions are the labels to be applied to the container
<code>seccompProfile</code>	<code>object</code>	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the <code>supplementalGroupsPolicy</code> 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 <code>supplementalGroupsPolicy</code> field. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when <code>spec.os.name</code> is windows. Possible enum values: <code>"Merge"</code> means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be merged with the primary user's groups as defined in the container image (in <code>/etc/group</code>).

Property	Type	Description
		"Strict" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be used instead of any groups defined in the container image.
sysctls	array	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

.spec.template.spec.securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

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. Possible enum values: "Localhost" indicates that a profile pre-loaded on the node should be used. "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. "Unconfined" indicates that no AppArmor profile should be enforced.

.spec.template.spec.securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.

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

`.spec.template.spec.securityContext.seccompProfile`

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied. Possible enum values: <code>"Localhost"</code> indicates a profile defined in a file on the node should be used. The file's location relative to <code>/seccomp</code>. <code>"RuntimeDefault"</code> represents the default container runtime seccomp profile. <code>"Unconfined"</code> indicates no seccomp profile is applied (A.K.A. unconfined).

`.spec.template.spec.securityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when `spec.os.name` is windows.

Type

`array`

.spec.template.spec.securityContext.supplementalGroups[]

Type

integer

.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

.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

.spec.template.spec.securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

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 <code>GMSACredentialSpecName</code> field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

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

`.spec.template.spec.tolerations`

Description

If specified, the pod's tolerations.

Type

`array`

`.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
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute. Possible enum values: <code>"NoExecute"</code> Evict any already-running pods that do not tolerate the taint. Currently enforced by NodeController. <code>"NoSchedule"</code> Do not allow new pods to schedule onto the node unless they tolerate the taint, but allow all pods submitted to Kubelet without going through the scheduler to start, and allow all already-running pods to continue running. Enforced by the scheduler. <code>"PreferNoSchedule"</code> Like TaintEffectNoSchedule, but the scheduler tries not to schedule new pods onto the node, rather than prohibiting new pods from scheduling onto the node entirely. Enforced by the scheduler.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.

Property	Type	Description
		Possible enum values: "Equal" "Exists"
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
value	string	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.template.spec.topologySpreadConstraints

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

array

.spec.template.spec.topologySpreadConstraints[]

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

object

Required

maxSkew topologyKey whenUnsatisfiable

Property	Type	Description
labelSelector	object	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.
matchLabelKeys	array	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 <code>whenUnsatisfiable=DoNotSchedule</code>, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P - if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1). - if MaxSkew is 2, incoming pod can be scheduled onto any zone. When <code>whenUnsatisfiable=ScheduleAnyway</code>, it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.
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.
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. Possible enum values: <code>"Honor"</code> means use this scheduling directive when calculating pod topology spread skew. <code>"Ignore"</code> means ignore this scheduling directive when calculating pod topology spread skew.
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. Possible enum values: <code>"Honor"</code> means use this scheduling directive when calculating pod topology spread skew. <code>"Ignore"</code> means ignore this scheduling directive when calculating pod topology spread skew.

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	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.
<code>whenUnsatisfiable</code>	<code>string</code>	WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. - DoNotSchedule (default) tells the scheduler not to schedule it. - ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: zone1 zone2 zone3 P P P P P If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it <i>more</i> imbalanced. It's a required field. Possible enum values: <code>"DoNotSchedule"</code> instructs the scheduler not to schedule the pod when constraints are not satisfied. <code>"ScheduleAnyway"</code> instructs the scheduler to schedule the pod even if constraints are not satisfied.

`.spec.template.spec.topologySpreadConstraints[].labelSelector`

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

`.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions`

Description

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

Type

`array`

.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[]

Description

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

Type

object

Required

key operator

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

.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]

Type

string

.spec.template.spec.topologySpreadConstraints[].labelSelector.matchLabels

Description

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

Type

object

.spec.template.spec.topologySpreadConstraints[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).

Type

array

`.spec.template.spec.topologySpreadConstraints[].matchLabelKeys[]`

Type

string

`.spec.template.spec.volumes`

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

`.spec.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
		Represents a Persistent Disk resource in AWS.
<code>awsElasticBlockStore</code>	<code>object</code>	An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.
<code>azureDisk</code>	<code>object</code>	AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.
<code>azureFile</code>	<code>object</code>	AzureFile represents an Azure File Service mount on the host and bind mount to the pod.
<code>cephfs</code>	<code>object</code>	Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.
<code>cinder</code>	<code>object</code>	Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership

Property	Type	Description
		management and SELinux relabeling.
<code>configMap</code>	<code>object</code>	Adapts a ConfigMap into a volume. The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.
<code>csi</code>	<code>object</code>	Represents a source location of a volume to mount, managed by an external CSI driver
<code>downwardAPI</code>	<code>object</code>	DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.
<code>emptyDir</code>	<code>object</code>	Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.
<code>ephemeral</code>	<code>object</code>	Represents an ephemeral volume that is handled by a normal storage driver.
<code>fc</code>	<code>object</code>	Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.
<code>flexVolume</code>	<code>object</code>	FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.
<code>flocker</code>	<code>object</code>	Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.
<code>gcePersistentDisk</code>	<code>object</code>	Represents a Persistent Disk resource in Google Compute Engine. A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.
<code>gitRepo</code>	<code>object</code>	Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling. 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.

Property	Type	Description
<code>glusterfs</code>	<code>object</code>	Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.
<code>hostPath</code>	<code>object</code>	Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.
<code>image</code>	<code>object</code>	ImageVolumeSource represents a image volume resource.
<code>iscsi</code>	<code>object</code>	Represents an ISCSI disk. ISCSI volumes can only be mounted as read/write once. ISCSI volumes support ownership management and SELinux relabeling.
<code>name</code>	<code>string</code>	name of the volume. Must be a DNS_LABEL and unique within the pod. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>nfs</code>	<code>object</code>	Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.
<code>persistentVolumeClaim</code>	<code>object</code>	PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).
<code>photonPersistentDisk</code>	<code>object</code>	Represents a Photon Controller persistent disk resource.
<code>portworxVolume</code>	<code>object</code>	PortworxVolumeSource represents a Portworx volume resource.
<code>projected</code>	<code>object</code>	Represents a projected volume source
<code>quobyte</code>	<code>object</code>	Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.
<code>rbd</code>	<code>object</code>	Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.
<code>scaleIO</code>	<code>object</code>	ScaleIOVolumeSource represents a persistent ScaleIO volume

Property	Type	Description
<code>secret</code>	<code>object</code>	Adapts a Secret into a volume. The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.
<code>storageos</code>	<code>object</code>	Represents a StorageOS persistent volume resource.
<code>vsphereVolume</code>	<code>object</code>	Represents a vSphere volume resource.

`.spec.template.spec.volumes[].awsElasticBlockStore`

Description

Represents a Persistent Disk resource in AWS. An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.

Type

`object`

Required

`volumeID`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
<code>partition</code>	<code>integer</code>	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).
<code>readOnly</code>	<code>boolean</code>	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
<code>volumeID</code>	<code>string</code>	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

`.spec.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
cacheMode	string	cacheMode is the Host Caching mode: None, Read Only, Read Write. Possible enum values: "None" "ReadOnly" "ReadWrite"
diskName	string	diskName is the Name of the data disk in the blob storage
diskURI	string	diskURI is the URI of data disk in the blob storage
fsType	string	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
kind	string	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared Possible enum values: "Dedicated" "Managed" "Shared"
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

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

`.spec.template.spec.volumes[].cephfs`

Description

Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.

Type

`object`

Required

`monitors`

Property	Type	Description
<code>monitors</code>	<code>array</code>	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>path</code>	<code>string</code>	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
<code>readOnly</code>	<code>boolean</code>	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 ↗
<code>secretFile</code>	<code>string</code>	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 ↗
<code>secretRef</code>	<code>object</code>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
<code>user</code>	<code>string</code>	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

`.spec.template.spec.volumes[].cephfs.monitors`

Description

monitors is Required: Monitors is a collection of Ceph monitors More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

array

.spec.template.spec.volumes[].cephfs.monitors[]

Type

string

.spec.template.spec.volumes[].cephfs.secretRef

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 ↗

.spec.template.spec.volumes[].cinder

Description

Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

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	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
volumeID	string	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

`.spec.template.spec.volumes[].cinder.secretRef`

Description

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

Type

object

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

`.spec.template.spec.volumes[].configMap`

Description

Adapts a ConfigMap into a volume. The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	optional specify whether the ConfigMap or its keys must be defined

`.spec.template.spec.volumes[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

.spec.template.spec.volumes[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

key path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.template.spec.volumes[].csi

Description

Represents a source location of a volume to mount, managed by an external CSI driver

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
<code>nodePublishSecretRef</code>	<code>object</code>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
<code>readOnly</code>	<code>boolean</code>	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
<code>volumeAttributes</code>	<code>object</code>	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

`.spec.template.spec.volumes[].csi.nodePublishSecretRef`

Description

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

Type

`object`

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

`.spec.template.spec.volumes[].csi.volumeAttributes`

Description

volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

`object`

`.spec.template.spec.volumes[].downwardAPI`

Description

DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.

Type

`object`

Property	Type	Description
defaultMode	integer	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	Items is a list of downward API volume file

.spec.template.spec.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

array

.spec.template.spec.volumes[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
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	ResourceFieldSelector represents container resources (cpu, memory) and their output format

.spec.template.spec.volumes[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

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

.spec.template.spec.volumes[].downwardAPI.items[].resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number less than 1e-30. When a Quantity is parsed from a string, it will remember the type of suffix it had, and will use that type when serializing. If no suffix is present, the quantity will be serialized as a decimal number. Before serializing, Quantity will be put in "canonical form". This means that Exponent notation will only be used if the exponent is not zero. - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be emitted if the number is not an integer The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. It is important to note that quantities with fractional digits are only allowed for decimal SI units. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form when serialized. This format is intended to make it difficult to use these numbers without writing software that understands the canonical format.
resource	string	Required: resource to select

.spec.template.spec.volumes[].emptyDir

Description

Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium.
sizeLimit	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and YAML. The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> <integer> ::= <digit> <digit><digits> <digit><digits>.<digits> <digit><digits>.<digits><decimalSI> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.html) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number When a Quantity is parsed from a string, it will remember the type of suffix it had, and Before serializing, Quantity will be put in "canonical form". This means that Exponent/Prefix - No precision is lost - No fractional digits will be emitted - The exponent (or suffix The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. Non-canonical values will still parse as long as they are well formed, but will be re-emitted This format is intended to make it difficult to use these numbers without writing some :

.spec.template.spec.volumes[].ephemeral

Description

Represents an ephemeral volume that is handled by a normal storage driver.

Type

object

Property	Type	Description
volumeClaimTemplate	object	PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate

Description

PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

Type

object

Required

spec

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

Description

PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.
dataSourceRef	object	TypedObjectReference contains enough information to let you locate the typed referenced object
resources	object	VolumeResourceRequirements describes the storage resource requirements for a volume.
selector	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.
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
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

Property	Type	Description
		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. Possible enum values: "Block" means the volume will not be formatted with a filesystem and will remain a raw block device. "Filesystem" means the volume will be or is formatted with a filesystem.
volumeName	string	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]

Type

string

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

Description

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

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

Property	Type	Description
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef`

Description

TypedObjectReference contains enough information to let you locate the typed referenced object

Type

`object`

Required

`kind` `name`

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced
<code>namespace</code>	<code>string</code>	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources`

Description

VolumeResourceRequirements describes the storage resource requirements for a volume.

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

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

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits

Description

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

Type

object

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests

Description

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

Type

object

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions

Description

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

Type

array

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]`

Description

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

Type

object

Required

key operator

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

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values`

Description

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

Type

array

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]`

Type

string

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels`

Description

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

Type

object

.spec.template.spec.volumes[].fc

Description

Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
lun	integer	lun is Optional: FC target lun number
readOnly	boolean	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
targetWWNs	array	targetWWNs is Optional: FC target worldwide names (WWNs)
wwids	array	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

.spec.template.spec.volumes[].fc.targetWWNs

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

array

.spec.template.spec.volumes[].fc.targetWWNs[]

Type

string

.spec.template.spec.volumes[].fc.wwids

Description

wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

Type

array

.spec.template.spec.volumes[].fc.wwids[]

Type

string

.spec.template.spec.volumes[].flexVolume

Description

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

Type

object

Required

driver

Property	Type	Description
driver	string	driver is the name of the driver to use for this volume.
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
options	object	options is Optional: this field holds extra command options if any.
readOnly	boolean	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

.spec.template.spec.volumes[].flexVolume.options

Description

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

Type

object

.spec.template.spec.volumes[].flexVolume.secretRef

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:

Property	Type	Description
https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗		

`.spec.template.spec.volumes[].flocker`

Description

Represents a Flocker volume mounted by the Flocker agent. One and only one of `datasetName` and `datasetUUID` should be set. Flocker volumes do not support ownership management or SELinux relabeling.

Type

object

Property	Type	Description
<code>datasetName</code>	string	<code>datasetName</code> is Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
<code>datasetUUID</code>	string	<code>datasetUUID</code> is the UUID of the dataset. This is unique identifier of a Flocker dataset

`.spec.template.spec.volumes[].gcePersistentDisk`

Description

Represents a Persistent Disk resource in Google Compute Engine. A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.

Type

object

Required

`pdName`

Property	Type	Description
<code>fsType</code>	string	<code>fsType</code> is filesystem type of the volume that you want to mount. 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 ↗
<code>partition</code>	integer	<code>partition</code> is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume <code>/dev/sda1</code> , you specify the partition as "1". Similarly, the volume partition for <code>/dev/sda</code> is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗
<code>pdName</code>	string	<code>pdName</code> is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> . Defaults to <code>false</code> . More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗

`.spec.template.spec.volumes[].gitRepo`

Description

Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling. DEPRECATED: `GitRepo` is deprecated. To provision a container with a git repo, mount an `EmptyDir` into an `InitContainer` that clones the repo using `git`, then mount the `EmptyDir` into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
<code>directory</code>	<code>string</code>	<code>directory</code> is the target directory name. Must not contain or start with <code>'..'</code> . If <code>'.'</code> is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
<code>repository</code>	<code>string</code>	<code>repository</code> is the URL
<code>revision</code>	<code>string</code>	<code>revision</code> is the commit hash for the specified revision.

`.spec.template.spec.volumes[].glusterfs`

Description

Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.

Type

`object`

Required

`endpoints` `path`

Property	Type	Description
<code>endpoints</code>	<code>string</code>	<code>endpoints</code> is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗
<code>path</code>	<code>string</code>	<code>path</code> is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗

`.spec.template.spec.volumes[].hostPath`

Description

Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗ Possible enum values: <code>""</code> For backwards compatible, leave it empty if unset <code>"BlockDevice"</code> A block device must exist at the given path <code>"CharDevice"</code> A character device must exist at the given path <code>"Directory"</code> A directory must exist at the given path <code>"DirectoryOrCreate"</code> If nothing exists at the given path, an empty directory will be created there as needed with file mode 0755, having the same group and ownership with Kubelet. <code>"File"</code> A file must exist at the given path <code>"FileOrCreate"</code> If nothing exists at the given path, an empty file will be created there as needed with file mode 0644, having the same group and ownership with Kubelet. <code>"Socket"</code> A UNIX socket must exist at the given path

`.spec.template.spec.volumes[].image`

Description

ImageVolumeSource represents a image volume resource.

Type

`object`

Property	Type	Description
<code>pullPolicy</code>	<code>string</code>	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact.

Property	Type	Description
		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. Possible enum values: • <code>"Always"</code> means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. • <code>"IfNotPresent"</code> means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. • <code>"Never"</code> means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
<code>reference</code>	<code>string</code>	Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images ^ This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

.spec.template.spec.volumes[].iscsi

Description

Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.

Type

`object`

Required

`targetPortal` `iqn` `lun`

Property	Type	Description
<code>chapAuthDiscovery</code>	<code>boolean</code>	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
<code>chapAuthSession</code>	<code>boolean</code>	chapAuthSession defines whether support iSCSI Session CHAP authentication
<code>fsType</code>	<code>string</code>	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi ^
<code>initiatorName</code>	<code>string</code>	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
<code>iqn</code>	<code>string</code>	iqn is the target iSCSI Qualified Name.

Property	Type	Description
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	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	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
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).

.spec.template.spec.volumes[].iscsi.portals

Description

portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

Type

array

.spec.template.spec.volumes[].iscsi.portals[]

Type

string

.spec.template.spec.volumes[].iscsi.secretRef

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

.spec.template.spec.volumes[].nfs

Description

Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.

Type

object

Required

server

path

Property	Type	Description
path	string	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ^
readOnly	boolean	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ^
server	string	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ^

.spec.template.spec.volumes[].persistentVolumeClaim

Description

PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).

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.

.spec.template.spec.volumes[].photonPersistentDisk

Description

Represents a Photon Controller persistent disk resource.

Type

object

Required

pdID

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
pdID	string	pdID is the ID that identifies Photon Controller persistent disk

.spec.template.spec.volumes[].portworxVolume

Description

PortworxVolumeSource represents a Portworx volume resource.

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fSType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

.spec.template.spec.volumes[].projected

Description

Represents a projected volume source

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.

Property	Type	Description
<code>sources</code>	<code>array</code>	<code>sources</code> is the list of volume projections. Each entry in this list handles one source.

`.spec.template.spec.volumes[].projected.sources`

Description

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

Type

`array`

`.spec.template.spec.volumes[].projected.sources[]`

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

`object`

Property	Type	Description
<code>clusterTrustBundle</code>	<code>object</code>	<code>ClusterTrustBundleProjection</code> describes how to select a set of <code>ClusterTrustBundle</code> objects and project their contents into the pod filesystem.
<code>configMap</code>	<code>object</code>	Adapts a <code>ConfigMap</code> into a projected volume. The contents of the target <code>ConfigMap</code>'s <code>Data</code> field will be presented in a projected volume as files using the keys in the <code>Data</code> field as the file names, unless the <code>items</code> element is populated with specific mappings of keys to paths. Note that this is identical to a <code>configmap</code> volume source without the default mode.
<code>downwardAPI</code>	<code>object</code>	Represents downward API info for projecting into a projected volume. Note that this is identical to a <code>downwardAPI</code> volume source without the default mode.
<code>secret</code>	<code>object</code>	Adapts a secret into a projected volume. The contents of the target <code>Secret</code>'s <code>Data</code> field will be presented in a projected volume as files using the keys in the <code>Data</code> field as the file names. Note that this is identical to a <code>secret</code> volume source without the default mode.
<code>serviceAccountToken</code>	<code>object</code>	<code>ServiceAccountTokenProjection</code> represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

`.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle`

Description

ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.

Type

object

Required

path

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

.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions

Description

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

Type

array

.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[]

Description

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

Type

object

Required

key operator

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

.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]

Type

string

.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchLabels

Description

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

Type

object

.spec.template.spec.volumes[].projected.sources[].configMap

Description

Adapts a ConfigMap into a projected volume. The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

.spec.template.spec.volumes[].projected.sources[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

.spec.template.spec.volumes[].projected.sources[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

key path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.template.spec.volumes[].projected.sources[].downwardAPI

Description

Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

Type

object

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

.spec.template.spec.volumes[].projected.sources[].downwardAPI.items

Description

Items is a list of DownwardAPIVolume file

Type

array

.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.

Property	Type	Description
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format

.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

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

.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and

Property	Type	Description
		The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number less than 10⁻³⁰. When a Quantity is parsed from a string, it will remember the type of suffix it had, and will use that type when serializing. For example, 1.5Gi will be serialized as "1536Mi". Before serializing, Quantity will be put in "canonical form". This means that Exponent notation will only be used if the decimal exponent is non-zero. - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be emitted if the decimal exponent is non-zero. The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. It is always represented as a string and parsed from that string. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form when serialized. This format is intended to make it difficult to use these numbers without writing software to parse and generate them.
resource	string	Required: resource to select

.spec.template.spec.volumes[].projected.sources[].secret

Description

Adapts a secret into a projected volume. The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.

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 '..'.

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 field specify whether the Secret or its key must be defined

.spec.template.spec.volumes[].projected.sources[].secret.items

Description

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

.spec.template.spec.volumes[].projected.sources[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

key path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.template.spec.volumes[].projected.sources[].serviceAccountToken

Description

ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

Type

object

Required

path

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours.Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

.spec.template.spec.volumes[].quobyte

Description

Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

Type

object

Required

registry volume

Property	Type	Description
group	string	group to map volume access to Default is no group
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user

Property	Type	Description
<code>volume</code>	<code>string</code>	volume is a string that references an already created Quobyte volume by name.

`.spec.template.spec.volumes[].rbd`

Description

Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.

Type

`object`

Required

`monitors``image`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd ↗
<code>image</code>	<code>string</code>	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
<code>keyring</code>	<code>string</code>	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 ↗
<code>monitors</code>	<code>array</code>	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
<code>pool</code>	<code>string</code>	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
<code>secretRef</code>	<code>object</code>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
<code>user</code>	<code>string</code>	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

`.spec.template.spec.volumes[].rbd.monitors`

Description

monitors is a collection of Ceph monitors. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

array

.spec.template.spec.volumes[].rbd.monitors[]

Type

string

.spec.template.spec.volumes[].rbd.secretRef

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

.spec.template.spec.volumes[].scaleIO

Description

ScaleIOVolumeSource represents a persistent ScaleIO volume

Type

object

Required

gateway system secretRef

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.

Property	Type	Description
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
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.

.spec.template.spec.volumes[].scaleIO.secretRef

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

.spec.template.spec.volumes[].secret

Description

Adapts a Secret into a volume. The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.

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

Property	Type	Description
		requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
optional	boolean	optional field specify whether the Secret or its keys must be defined
secretName	string	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

.spec.template.spec.volumes[].secret.items

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

.spec.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.

Property	Type	Description
<code>path</code>	<code>string</code>	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.template.spec.volumes[].storageos`

Description

Represents a StorageOS persistent volume resource.

Type

`object`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>readOnly</code>	<code>boolean</code>	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>secretRef</code>	<code>object</code>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
<code>volumeName</code>	<code>string</code>	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
<code>volumeNamespace</code>	<code>string</code>	volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

`.spec.template.spec.volumes[].storageos.secretRef`

Description

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

Type

`object`

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

`.spec.template.spec.volumes[].vsphereVolume`

Description

Represents a vSphere volume resource.

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

`.spec.updateStrategy`

Description

DaemonSetUpdateStrategy is a struct used to control the update strategy for a DaemonSet.

Type

object

Property	Type	Description
rollingUpdate	object	Spec to control the desired behavior of daemon set rolling update.
type	string	Type of daemon set update. Can be "RollingUpdate" or "OnDelete". Default is RollingUpdate. Possible enum values: "OnDelete" Replace the old daemons only when it's killed "RollingUpdate" Replace the old daemons by new ones using rolling update i.e replace them on each node one after the other.

`.spec.updateStrategy.rollingUpdate`

Description

Spec to control the desired behavior of daemon set rolling update.

Type

object

Property	Type	Description
maxSurge	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
maxUnavailable	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.status

Description

DaemonSetStatus represents the current status of a daemon set.

Type

object

Required

currentNumberScheduled numberMisscheduled desiredNumberScheduled numberReady

Property	Type	Description
collisionCount	integer	Count of hash collisions for the DaemonSet. The DaemonSet controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ControllerRevision.
conditions	array	Represents the latest available observations of a DaemonSet's current state.
currentNumberScheduled	integer	The number of nodes that are running at least 1 daemon pod and are supposed to run the daemon pod. More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
desiredNumberScheduled	integer	The total number of nodes that should be running the daemon pod (including nodes correctly running the daemon pod). More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
numberAvailable	integer	The number of nodes that should be running the daemon pod and have one or more of the daemon pod running and available (ready for at least spec.minReadySeconds)
numberMisscheduled	integer	The number of nodes that are running the daemon pod, but are not supposed to run the daemon pod. More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/

Property	Type	Description
<code>numberReady</code>	<code>integer</code>	<code>numberReady</code> is the number of nodes that should be running the daemon pod and have one or more of the daemon pod running with a Ready Condition.
<code>numberUnavail able</code>	<code>integer</code>	The number of nodes that should be running the daemon pod and have none of the daemon pod running and available (ready for at least <code>spec.minReadySeconds</code>)
<code>observedGeneration</code>	<code>integer</code>	The most recent generation observed by the daemon set controller.
<code>updatedNumberScheduled</code>	<code>integer</code>	The total number of nodes that are running updated daemon pod

`.status.conditions`

Description

Represents the latest available observations of a DaemonSet's current state.

Type

`array`

`.status.conditions[]`

Description

DaemonSetCondition describes the state of a DaemonSet at a certain point.

Type

`object`

Required

`type` `status`

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.
<code>message</code>	<code>string</code>	A human readable message indicating details about the transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.

Property	Type	Description
type	string	Type of DaemonSet condition.

API Endpoints

The following API endpoints are available:

- `/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/daemonsets`
 - `DELETE` : delete collection of DaemonSet
 - `GET` : list objects of kind DaemonSet
 - `POST` : create a new DaemonSet
- `/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}`
 - `DELETE` : delete the specified DaemonSet
 - `GET` : read the specified DaemonSet
 - `PATCH` : partially update the specified DaemonSet
 - `PUT` : replace the specified DaemonSet
- `/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status`
 - `GET` : read status of the specified DaemonSet
 - `PATCH` : partially update status of the specified DaemonSet
 - `PUT` : replace status of the specified DaemonSet

/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/daemonsets

HTTP method

DELETE

Description

delete collection of DaemonSet

HTTP responses

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

HTTP method

GET

Description

list objects of kind DaemonSet

HTTP responses

HTTP code	Response body
200 - OK	DaemonSetList schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new DaemonSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>DaemonSet</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>DaemonSet</code> schema
201 - Created	<code>DaemonSet</code> schema
202 - Accepted	<code>DaemonSet</code> schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}**HTTP method**

DELETE

Description

delete the specified DaemonSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified DaemonSet

HTTP responses

HTTP code	Response body
200 - OK	<code>DaemonSet</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified DaemonSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>DaemonSet</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified DaemonSet

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	DaemonSet schema	application/json formatted

HTTP responses

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

/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status

HTTP method

GET

Description

read status of the specified DaemonSet

HTTP responses

HTTP code	Response body
200 - OK	DaemonSet schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified DaemonSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>DaemonSet</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified DaemonSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>DaemonSet</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>DaemonSet</code> schema
201 - Created	<code>DaemonSet</code> schema
401 - Unauthorized	Empty

Deployments [apps/v1]

Description

Deployment enables declarative updates for Pods and ReplicaSets.

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	DeploymentSpec is the specification of the desired behavior of the Deployment.
<code>status</code>	<code>object</code>	DeploymentStatus is the most recently observed status of the Deployment.

.spec

Description

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

Type

object

Required

selector template

Property	Type	Description
<code>minReadySeconds</code>	<code>integer</code>	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)
<code>paused</code>	<code>boolean</code>	Indicates that the deployment is paused.
<code>progressDeadlineSeconds</code>	<code>integer</code>	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.
<code>replicas</code>	<code>integer</code>	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
<code>revisionHistoryLimit</code>	<code>integer</code>	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.
<code>selector</code>	<code>object</code>	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.
<code>strategy</code>	<code>object</code>	DeploymentStrategy describes how to replace existing pods with new ones.
<code>template</code>	<code>object</code>	PodTemplateSpec describes the data a pod should have when created from a template

.spec.selector

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

Property	Type	Description
		"value". The requirements are ANDed.

.spec.selector.matchExpressions

Description

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

Type

array

.spec.selector.matchExpressions[]

Description

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

Type

object

Required

key operator

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

.spec.selector.matchExpressions[].values

Description

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

Type

array

.spec.selector.matchExpressions[].values[]

Type

string

.spec.selector.matchLabels

Description

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

Type

object

.spec.strategy

Description

DeploymentStrategy describes how to replace existing pods with new ones.

Type

object

Property	Type	Description
rollingUpdate	object	Spec to control the desired behavior of rolling update.
type	string	Type of deployment. Can be "Recreate" or "RollingUpdate". Default is RollingUpdate. Possible enum values: "Recreate" Kill all existing pods before creating new ones. "RollingUpdate" Replace the old ReplicaSets by new one using rolling update i.e gradually scale down the old ReplicaSets and scale up the new one.

.spec.strategy.rollingUpdate

Description

Spec to control the desired behavior of rolling update.

Type

object

Property	Type	Description
maxSurge	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
maxUnavailable	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template

Description

PodTemplateSpec describes the data a pod should have when created from a template

Type

object

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	PodSpec is a description of a pod.

.spec.template.spec

Description

PodSpec is a description of a pod.

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	Affinity is a group of affinity scheduling rules.
automountServiceAccountToken	boolean	AutomountServiceAccountToken indicates whether a service account token should be automatically mounted.
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.
dnsConfig	object	PodDNSConfig defines the DNS parameters of a pod in addition to those generated from 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'. Possible enum values: "ClusterFirst" indicates that the pod should use cluster DNS first unless hostNetwork is true, if it is available, then fall back on the default (as determined by kubelet) DNS settings.

Property	Type	Description
		"<code>ClusterFirstWithHostNet</code>" indicates that the pod should use cluster DNS first, if it is available, then fall back on the default (as determined by kubelet) DNS settings. "<code>Default</code>" indicates that the pod should use the default (as determined by kubelet) DNS settings. "<code>None</code>" indicates that the pod should use empty DNS settings. DNS parameters such as nameservers and search paths should be defined via DNSConfig.
<code>enableServiceLinks</code>	<code>boolean</code>	EnableServiceLinks indicates whether information about services should be injected into pod's environment variables, matching the syntax of Docker links. Optional: Defaults to true.
<code>ephemeralContainers</code>	<code>array</code>	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.
<code>hostAliases</code>	<code>array</code>	HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
<code>hostIPC</code>	<code>boolean</code>	Use the host's ipc namespace. Optional: Default to false.
<code>hostNetwork</code>	<code>boolean</code>	Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified. Default to false.
<code>hostPID</code>	<code>boolean</code>	Use the host's pid namespace. Optional: Default to false.
<code>hostUsers</code>	<code>boolean</code>	Use the host's user namespace. Optional: Default to 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.
<code>hostname</code>	<code>string</code>	Specifies the hostname of the Pod If not specified, the pod's hostname will be set to a system-defined value.
<code>imagePullSecrets</code>	<code>array</code>	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

Property	Type	Description
<code>initContainers</code>	<code>array</code>	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/ ↗
<code>nodeName</code>	<code>string</code>	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/docs/concepts/scheduling-eviction/assign-pod-node/#nodename ↗
<code>nodeSelector</code>	<code>object</code>	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/ ↗
<code>os</code>	<code>object</code>	PodOS defines the OS parameters of a pod.
<code>overhead</code>	<code>object</code>	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 ↗
<code>preemptionPolicy</code>	<code>string</code>	PreemptionPolicy is the Policy for preempting pods with lower priority. One of Never, PreemptLowerPriority. Defaults to PreemptLowerPriority if unset. Possible enum values: <code>"Never"</code> means that pod never preempts other pods with lower priority. <code>"PreemptLowerPriority"</code> means that pod can preempt other pods with lower priority.
<code>priority</code>	<code>integer</code>	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
<code>priorityClassName</code>	<code>string</code>	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.
<code>readinessGates</code>	<code>array</code>	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 ↗
<code>resourceClaims</code>	<code>array</code>	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.
<code>resources</code>	<code>object</code>	ResourceRequirements describes the compute resource requirements.
<code>restartPolicy</code>	<code>string</code>	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/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy ↗ Possible enum values: <code>"Always"</code> <code>"Never"</code> <code>"OnFailure"</code>
<code>runtimeClassName</code>	<code>string</code>	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 ↗
<code>schedulerName</code>	<code>string</code>	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
<code>schedulingGates</code>	<code>array</code>	SchedulingGates is an opaque list of values that 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.

Property	Type	Description
<code>securityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
<code>serviceAccount</code>	<code>string</code>	DeprecatedServiceAccount is a deprecated alias for ServiceAccountName. Deprecated: Use serviceAccountName instead.
<code>serviceAccountName</code>	<code>string</code>	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/ ^
<code>setHostnameAsFQDN</code>	<code>boolean</code>	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.
<code>shareProcessNamespace</code>	<code>boolean</code>	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.
<code>subdomain</code>	<code>string</code>	If specified, the fully qualified Pod hostname will be "...svc.". If not specified, the pod will not have a domainname at all.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no 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.
<code>tolerations</code>	<code>array</code>	If specified, the pod's tolerations.
<code>topologySpreadConstraints</code>	<code>array</code>	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.

Property	Type	Description
<code>volumes</code>	<code>array</code>	List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes ^

.spec.template.spec.affinity

Description

Affinity is a group of affinity scheduling rules.

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Node affinity is a group of node affinity scheduling rules.
<code>podAffinity</code>	<code>object</code>	Pod affinity is a group of inter pod affinity scheduling rules.
<code>podAntiAffinity</code>	<code>object</code>	Pod anti affinity is a group of inter pod anti affinity scheduling rules.

.spec.template.spec.affinity.nodeAffinity

Description

Node affinity is a group of node affinity scheduling rules.

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, 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.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

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

Description

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

Type

object

Required

weight preference

Property	Type	Description
preference	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.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

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

Description

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

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Description

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

Type

object

Required

key

operator

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

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

Description

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

Type

array

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

Type

string

.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

Type

object

Required

nodeSelectorTerms

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

.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

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

Type

array

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

Description

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

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

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

Description

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

Type

array

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

Type

string

.spec.template.spec.affinity.podAffinity

Description

Pod affinity is a group of inter pod affinity scheduling rules.

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

Property	Type	Description
		corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
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.

.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

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

Description

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

Type

object

Required

weight podAffinityTerm

Property	Type	Description
podAffinityTerm	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 matches that of any node on which a pod of the set of pods is running
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

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

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 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.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key not 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 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 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.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

keyoperator

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

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

Description

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

Type

`array`

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

Type

`string`

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

Description

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

Type

`object`

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

Description

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

Type

`array`

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

Type

`string`

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

Description

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

Type

array

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

Type

string

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key

operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

`.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

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

Description

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

Type

object

Required

topologyKey

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

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

.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution.labelSelector

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution.labelSelector.matchExpressions

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

`array`

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

Type

`string`

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

Description

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

Type

`array`

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

Type

`string`

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

`object`

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

`.spec.template.spec.affinity.podAntiAffinity`

Description

Pod anti affinity is a group of inter pod anti affinity scheduling rules.

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	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

Property	Type	Description
		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.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	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.

.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

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

Description

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

Type

`object`

Required

`weight` `podAffinityTerm`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	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 matches that of any node on which a pod of the set of pods is running
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

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

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 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.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key not 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 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 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.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Type

string

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key

operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

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

Description

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

Type

object

Required

topologyKey

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

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

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

array

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

Type

string

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

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

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

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

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Description

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

Type

array

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

Type

string

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

Description

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

Type

object

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

Description

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

Type

array

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

Type

string

.spec.template.spec.containers

Description

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

Type

array

.spec.template.spec.containers[]

Description

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

Type

object

Required

Property	Type	Description
args	array	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must 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.
image	string	Container image name. 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.
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: "Always" means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present

Property	Type	Description
<code>lifecycle</code>	<code>object</code>	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
<code>livenessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>name</code>	<code>string</code>	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
<code>ports</code>	<code>array</code>	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. 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.
<code>readinessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	ResourceRequirements describes the compute resource requirements.
<code>restartPolicy</code>	<code>string</code>	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers 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.
<code>securityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>startupProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
<code>stdin</code>	<code>boolean</code>	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.
<code>stdinOnce</code>	<code>boolean</code>	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
<code>terminationMessagePath</code>	<code>string</code>	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.
<code>terminationMessagePolicy</code>	<code>string</code>	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. Possible enum values: <code>"FallbackToLogsOnError"</code> will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. <code>"File"</code> is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
<code>volumeDevices</code>	<code>array</code>	volumeDevices is the list of block devices to be used by the container.
<code>volumeMounts</code>	<code>array</code>	Pod volumes to mount into the container's filesystem. Cannot be updated.
<code>workingDir</code>	<code>string</code>	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

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

Description

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

Type

array

.spec.template.spec.containers[].args[]**Type**

string

.spec.template.spec.containers[].command**Description**

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

Type

array

.spec.template.spec.containers[].command[]**Type**

string

.spec.template.spec.containers[].env**Description**

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

Type

array

.spec.template.spec.containers[].env[]**Description**

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

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

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

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

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

Description

Selects a key from a ConfigMap.

Type

object

Required

key

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

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

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

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

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

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number. When a Quantity is parsed from a string, it will remember the type of suffix it had, and the exponent. Before serializing, Quantity will be put in "canonical form". This means that Exponent will be normalized. - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be normalized The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form. This format is intended to make it difficult to use these numbers without writing something like
resource	string	Required: resource to select

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

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.

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

.spec.template.spec.containers[].envFrom

Description

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

Type

array

.spec.template.spec.containers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

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

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

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

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

Description

SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.template.spec.containers[].lifecycle`

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
<code>preStop</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

`.spec.template.spec.containers[].lifecycle.postStart`

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

.spec.template.spec.containers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.containers[].lifecycle.postStart.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.containers[].lifecycle.postStart.exec.command[]

Type

string

.spec.template.spec.containers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.containers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.template.spec.containers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.containers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

.spec.template.spec.containers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ("/) in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (!, etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.containers[].lifecycle.preStop.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ("/) in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (!, etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.containers[].lifecycle.preStop.exec.command[]

Type

string

.spec.template.spec.containers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.containers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.template.spec.containers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.containers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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	TCPSocketAction describes an action based on opening a socket
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

.spec.template.spec.containers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.containers[].livenessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.containers[].livenessProbe.exec.command[]

Type

string

.spec.template.spec.containers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

.spec.template.spec.containers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.

Property	Type	Description
value	string	The header field value

.spec.template.spec.containers[].livenessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.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

.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.

Property	Type	Description
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: "SCTP" is the SCTP protocol. "TCP" is the TCP protocol. "UDP" is the UDP protocol.

.spec.template.spec.containers[].readinessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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	TCPSocketAction describes an action based on opening a socket
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

`.spec.template.spec.containers[].readinessProbe.exec`

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (!', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.template.spec.containers[].readinessProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (!', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.template.spec.containers[].readinessProbe.exec.command[]`

Type

string

.spec.template.spec.containers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

.spec.template.spec.containers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

Property	Type	Description
		Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.containers[].readinessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.containers[].resizePolicy

Description

Resources resize policy for the container.

Type

`array`

.spec.template.spec.containers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

.spec.template.spec.containers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
<code>claims</code>	<code>array</code>	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Property	Type	Description
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.template.spec.containers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.template.spec.containers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.template.spec.containers[].resources.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.containers[].resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.containers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

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 defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.
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. Possible enum values: "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	SELinuxOptions are the labels to be applied to the container
<code>seccompProfile</code>	<code>object</code>	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
<code>windowsOptions</code>	<code>object</code>	WindowsSecurityContextOptions contain Windows-specific options and credentials.

`.spec.template.spec.containers[].securityContext.appArmorProfile`

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: <code>"Localhost"</code> indicates that a profile pre-loaded on the node should be used.

Property	Type	Description
		"<code>RuntimeDefault</code>" indicates that the container runtime's default AppArmor profile should be used. "<code>Unconfined</code>" indicates that no AppArmor profile should be enforced.

.spec.template.spec.containers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array	Added capabilities
drop	array	Removed capabilities

.spec.template.spec.containers[].securityContext.capabilities.add

Description

Added capabilities

Type

array

.spec.template.spec.containers[].securityContext.capabilities.add[]

Type

string

.spec.template.spec.containers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

array

.spec.template.spec.containers[].securityContext.capabilities.drop[]

Type

string

.spec.template.spec.containers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.template.spec.containers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

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. Possible enum values: "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to /seccomp. "RuntimeDefault" represents the default container runtime seccomp profile. "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

.spec.template.spec.containers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

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.

.spec.template.spec.containers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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 ↗

Property	Type	Description
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	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.
<code>tcpSocket</code>	<code>object</code>	TCPSocketAction describes an action based on opening a socket
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	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.
<code>timeoutSeconds</code>	<code>integer</code>	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

`.spec.template.spec.containers[].startupProbe.exec`

Description

ExecAction describes a "run in container" action.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.template.spec.containers[].startupProbe.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.template.spec.containers[].startupProbe.exec.command[]

Type

string

.spec.template.spec.containers[].startupProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ^). If this is not specified, the default behavior is defined by gRPC.

.spec.template.spec.containers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

Property	Type	Description
		Scheme to use for connecting to the host. Defaults to HTTP.
scheme	string	Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.containers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.containers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.containers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.containers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

.spec.template.spec.containers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

name devicePath

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

.spec.template.spec.containers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

array

.spec.template.spec.containers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

name mountPath

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). Possible enum values: "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
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).
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.

Description

PodDNSConfig defines the DNS parameters of a pod in addition to those generated from DNSPolicy.

Type

object

Property	Type	Description
nameservers	array	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
options	array	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.
searches	array	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

.spec.template.spec.dnsConfig.nameservers

Description

A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.

Type

array

.spec.template.spec.dnsConfig.nameservers[]

Type

string

.spec.template.spec.dnsConfig.options

Description

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type

array

.spec.template.spec.dnsConfig.options[]

Description

PodDNSConfigOption defines DNS resolver options of a pod.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name is this DNS resolver option's name. Required.
<code>value</code>	<code>string</code>	Value is this DNS resolver option's value.

.spec.template.spec.dnsConfig.searches

Description

A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

Type

`array`

.spec.template.spec.dnsConfig.searches[]

Type

`string`

.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`

.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
<code>args</code>	<code>array</code>	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

Property	Type	Description
		exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell ↗
command	array	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(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.
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.
image	string	Container image name. More info: 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/concepts/containers/images#updating-images ↗ Possible enum values: "Always" means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
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.

Property	Type	Description
<code>ports</code>	<code>array</code>	Ports are not allowed for ephemeral containers.
<code>readinessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	ResourceRequirements describes the compute resource requirements.
<code>restartPolicy</code>	<code>string</code>	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.
<code>securityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>startupProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>stdin</code>	<code>boolean</code>	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.
<code>stdinOnce</code>	<code>boolean</code>	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
<code>targetContainerName</code>	<code>string</code>	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.
<code>terminationMessagePath</code>	<code>string</code>	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

Property	Type	Description
		length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.
		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.
terminationMessagePolicy	string	Possible enum values: "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
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.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.
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.

.spec.template.spec.ephemeralContainers[].args

Description

Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.template.spec.ephemeralContainers[].args[]

Type

string

.spec.template.spec.ephemeralContainers[].command

Description

Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.template.spec.ephemeralContainers[].command[]

Type

string

.spec.template.spec.ephemeralContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

.spec.template.spec.ephemeralContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

.spec.template.spec.ephemeralContainers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

.spec.template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type

object

Required

key

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

.spec.template.spec.ephemeralContainers[].env[].valueFrom.fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

.spec.template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number with a precision greater than 16 digits. When a Quantity is parsed from a string, it will remember the type of suffix it had, and will use that type when serializing. Before serializing, Quantity will be put in "canonical form". This means that Exponent and Suffix will be normalized to the following: - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. All internal calculations are done using integers. Non-canonical values will still parse as long as they are well formed, but will be rounded to the canonical form when serialized. This format is intended to make it difficult to use these numbers without writing software that understands the format.
resource	string	Required: resource to select

.spec.template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
optional	boolean	Specify whether the Secret or its key must be defined

.spec.template.spec.ephemeralContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

.spec.template.spec.ephemeralContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

.spec.template.spec.ephemeralContainers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap must be defined

`.spec.template.spec.ephemeralContainers[].envFrom[].secretRef`

Description

SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.template.spec.ephemeralContainers[].lifecycle`

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
<code>preStop</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

`.spec.template.spec.ephemeralContainers[].lifecycle.postStart`

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec.command[]

Type

string

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.ephemeralContainers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

.spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (, etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (|, etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec.command[]

Type

string

.spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.ephemeralContainers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.ephemeralContainers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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	TCPSocketAction describes an action based on opening a socket
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

`.spec.template.spec.ephemeralContainers[].livenessProbe.exec`

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.ephemeralContainers[].livenessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.ephemeralContainers[].livenessProbe.exec.command[]

Type

string

.spec.template.spec.ephemeralContainers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

.spec.template.spec.ephemeralContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.

Property	Type	Description
value	string	The header field value

.spec.template.spec.ephemeralContainers[].livenessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.ephemeralContainers[].ports

Description

Ports are not allowed for ephemeral containers.

Type

array

.spec.template.spec.ephemeralContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

containerPort

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.
hostIP	string	What host IP to bind the external port to.

Property	Type	Description
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". Possible enum values: "SCTP" is the SCTP protocol. "TCP" is the TCP protocol. "UDP" is the UDP protocol.

.spec.template.spec.ephemeralContainers[].readinessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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	TCPSocketAction describes an action based on opening a socket
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

.spec.template.spec.ephemeralContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.ephemeralContainers[].readinessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.ephemeralContainers[].readinessProbe.exec.command[]

Type

string

.spec.template.spec.ephemeralContainers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

.spec.template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

Property	Type	Description
		Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.ephemeralContainers[].readinessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.ephemeralContainers[].resizePolicy

Description

Resources resize policy for the container.

Type

`array`

.spec.template.spec.ephemeralContainers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

.spec.template.spec.ephemeralContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
<code>claims</code>	<code>array</code>	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.template.spec.ephemeralContainers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.template.spec.ephemeralContainers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.template.spec.ephemeralContainers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.template.spec.ephemeralContainers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.ephemeralContainers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

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 defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.
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. Possible enum values: "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

.spec.template.spec.ephemeralContainers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

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. Possible enum values: "Localhost" indicates that a profile pre-loaded on the node should be used.

Property	Type	Description
		"<code>RuntimeDefault</code>" indicates that the container runtime's default AppArmor profile should be used. "<code>Unconfined</code>" indicates that no AppArmor profile should be enforced.

.spec.template.spec.ephemeralContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array	Added capabilities
drop	array	Removed capabilities

.spec.template.spec.ephemeralContainers[].securityContext.capabilities.add

Description

Added capabilities

Type

array

.spec.template.spec.ephemeralContainers[].securityContext.capabilities.add[]

Type

string

.spec.template.spec.ephemeralContainers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

array

.spec.template.spec.ephemeralContainers[].securityContext.capabilities.drop[]

Type

string

.spec.template.spec.ephemeralContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.template.spec.ephemeralContainers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

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. Possible enum values: "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to /seccomp. "RuntimeDefault" represents the default container runtime seccomp profile. "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

.spec.template.spec.ephemeralContainers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

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.

.spec.template.spec.ephemeralContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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 ↗

Property	Type	Description
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	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.
<code>tcpSocket</code>	<code>object</code>	TCPSocketAction describes an action based on opening a socket
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	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.
<code>timeoutSeconds</code>	<code>integer</code>	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

.spec.template.spec.ephemeralContainers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.ephemeralContainers[].startupProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.template.spec.ephemeralContainers[].startupProbe.exec.command[]`

Type

string

`.spec.template.spec.ephemeralContainers[].startupProbe.grpc`

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ^). If this is not specified, the default behavior is defined by gRPC.

`.spec.template.spec.ephemeralContainers[].startupProbe.httpGet`

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

Property	Type	Description
		Scheme to use for connecting to the host. Defaults to HTTP.
scheme	string	Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.ephemeralContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.ephemeralContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

`array`

.spec.template.spec.ephemeralContainers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

`object`

Required

`name` `devicePath`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	devicePath is the path inside of the container that the device will be mapped to.
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

.spec.template.spec.ephemeralContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

`array`

.spec.template.spec.ephemeralContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`name` `mountPath`

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). Possible enum values: "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
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).
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.

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

array

`.spec.template.spec.hostAliases[]`

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type

object

Required

ip

Property	Type	Description
<code>hostnames</code>	<code>array</code>	Hostnames for the above IP address.
<code>ip</code>	<code>string</code>	IP address of the host file entry.

`.spec.template.spec.hostAliases[].hostnames`

Description

Hostnames for the above IP address.

Type

array

`.spec.template.spec.hostAliases[].hostnames[]`

Type

string

`.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

`.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 ↗

.spec.template.spec.initContainers

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

Type

array

.spec.template.spec.initContainers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

name

Property	Type	Description
args	array	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" 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
<code>command</code>	<code>array</code>	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
<code>env</code>	<code>array</code>	List of environment variables to set in the container. Cannot be updated.
<code>envFrom</code>	<code>array</code>	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.
<code>image</code>	<code>string</code>	Container image name. 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.
<code>imagePullPolicy</code>	<code>string</code>	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: <code>"Always"</code> means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. <code>"IfNotPresent"</code> means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. <code>"Never"</code> means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
<code>lifecycle</code>	<code>object</code>	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
<code>livenessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>name</code>	<code>string</code>	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.

Property	Type	Description
<code>ports</code>	<code>array</code>	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. 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.
<code>readinessProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>resizePolicy</code>	<code>array</code>	Resources resize policy for the container.
<code>resources</code>	<code>object</code>	ResourceRequirements describes the compute resource requirements.
<code>restartPolicy</code>	<code>string</code>	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers 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.
<code>securityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>startupProbe</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>stdin</code>	<code>boolean</code>	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.
<code>stdinOnce</code>	<code>boolean</code>	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. Possible enum values: "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
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.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Cannot be updated.
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.

.spec.template.spec.initContainers[].args

Description

Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.template.spec.initContainers[].args[]

Type

string

.spec.template.spec.initContainers[].command

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

Type

array

.spec.template.spec.initContainers[].command[]

Type

string

.spec.template.spec.initContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

.spec.template.spec.initContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double \$\$ are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

.spec.template.spec.initContainers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

.spec.template.spec.initContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type

object

Required

key

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

.spec.template.spec.initContainers[].env[].valueFrom.fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

.spec.template.spec.initContainers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number. When a Quantity is parsed from a string, it will remember the type of suffix it had, and will use that type when serializing. Before serializing, Quantity will be put in "canonical form". This means that Exponent and Suffix will be normalized in the following ways: - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be as small as possible. The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form when serialized. This format is intended to make it difficult to use these numbers without writing something like:
resource	string	Required: resource to select

.spec.template.spec.initContainers[].env[].valueFrom.secretKeyRef

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

.spec.template.spec.initContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

.spec.template.spec.initContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

.spec.template.spec.initContainers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap must be defined

`.spec.template.spec.initContainers[].envFrom[].secretRef`

Description

SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret must be defined

`.spec.template.spec.initContainers[].lifecycle`

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

`object`

Property	Type	Description
<code>postStart</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
<code>preStop</code>	<code>object</code>	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

`.spec.template.spec.initContainers[].lifecycle.postStart`

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
<code>exec</code>	object	ExecAction describes a "run in container" action.
<code>httpGet</code>	object	HTTPGetAction describes an action based on HTTP Get requests.
<code>sleep</code>	object	SleepAction describes a "sleep" action.
<code>tcpSocket</code>	object	TCPSocketAction describes an action based on opening a socket

`.spec.template.spec.initContainers[].lifecycle.postStart.exec`

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
<code>command</code>	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.template.spec.initContainers[].lifecycle.postStart.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.template.spec.initContainers[].lifecycle.postStart.exec.command[]`

Type

string

`.spec.template.spec.initContainers[].lifecycle.postStart.httpGet`

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

`.spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

`.spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.initContainers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.initContainers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

.spec.template.spec.initContainers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.initContainers[].lifecycle.preStop.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.initContainers[].lifecycle.preStop.exec.command[]

Type

string

.spec.template.spec.initContainers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.initContainers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

seconds

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

.spec.template.spec.initContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.initContainers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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	TCPSocketAction describes an action based on opening a socket
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

.spec.template.spec.initContainers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.initContainers[].livenessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.initContainers[].livenessProbe.exec.command[]

Type

string

.spec.template.spec.initContainers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

.spec.template.spec.initContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.

Property	Type	Description
value	string	The header field value

.spec.template.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.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

.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.

Property	Type	Description
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: <code>"SCTP"</code> is the SCTP protocol. <code>"TCP"</code> is the TCP protocol. <code>"UDP"</code> is the UDP protocol.

`.spec.template.spec.initContainers[].readinessProbe`

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	ExecAction describes a "run in container" action.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPCAction specifies an action involving a GRPC service.
<code>httpGet</code>	<code>object</code>	HTTPGetAction describes an action based on HTTP Get requests.
<code>initialDelaySeconds</code>	<code>integer</code>	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
<code>periodSeconds</code>	<code>integer</code>	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	TCPSocketAction describes an action based on opening a socket
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

.spec.template.spec.initContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (!', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.initContainers[].readinessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (!', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.template.spec.initContainers[].readinessProbe.exec.command[]

Type

string

.spec.template.spec.initContainers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

.spec.template.spec.initContainers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP.

Property	Type	Description
		Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.initContainers[].readinessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.initContainers[].resizePolicy

Description

Resources resize policy for the container.

Type

`array`

.spec.template.spec.initContainers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

Property	Type	Description
<code>resourceName</code>	<code>string</code>	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
<code>restartPolicy</code>	<code>string</code>	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

.spec.template.spec.initContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
<code>claims</code>	<code>array</code>	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.template.spec.initContainers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.template.spec.initContainers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.template.spec.initContainers[].resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.template.spec.initContainers[].resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.initContainers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the 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.
<code>appArmorProfile</code>	<code>object</code>	AppArmorProfile defines a pod or container's AppArmor settings.
<code>capabilities</code>	<code>object</code>	Adds and removes POSIX capabilities from running containers.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
<code>procMount</code>	<code>string</code>	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows. Possible enum values: <code>"Default"</code> uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. <code>"Unmasked"</code> bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

.spec.template.spec.initContainers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

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. Possible enum values: "Localhost" indicates that a profile pre-loaded on the node should be used.

Property	Type	Description
		"<code>RuntimeDefault</code>" indicates that the container runtime's default AppArmor profile should be used. "<code>Unconfined</code>" indicates that no AppArmor profile should be enforced.

.spec.template.spec.initContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array	Added capabilities
drop	array	Removed capabilities

.spec.template.spec.initContainers[].securityContext.capabilities.add

Description

Added capabilities

Type

array

.spec.template.spec.initContainers[].securityContext.capabilities.add[]

Type

string

.spec.template.spec.initContainers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

array

.spec.template.spec.initContainers[].securityContext.capabilities.drop[]

Type

string

.spec.template.spec.initContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.template.spec.initContainers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

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. Possible enum values: "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to /seccomp. "RuntimeDefault" represents the default container runtime seccomp profile. "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

.spec.template.spec.initContainers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

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.

.spec.template.spec.initContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
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 ↗

Property	Type	Description
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	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.
<code>tcpSocket</code>	<code>object</code>	TCPSocketAction describes an action based on opening a socket
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	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.
<code>timeoutSeconds</code>	<code>integer</code>	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

.spec.template.spec.initContainers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.template.spec.initContainers[].startupProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.template.spec.initContainers[].startupProbe.exec.command[]`

Type

string

`.spec.template.spec.initContainers[].startupProbe.grpc`

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ^). If this is not specified, the default behavior is defined by gRPC.

`.spec.template.spec.initContainers[].startupProbe.httpGet`

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port	integer string	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

Property	Type	Description
		Scheme to use for connecting to the host. Defaults to HTTP.
scheme	string	Possible enum values: "HTTP" means that the scheme used will be http:// "HTTPS" means that the scheme used will be https://

.spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

.spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer string</code>	IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

.spec.template.spec.initContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

`array`

.spec.template.spec.initContainers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

`object`

Required

`name` `devicePath`

Property	Type	Description
<code>devicePath</code>	<code>string</code>	devicePath is the path inside of the container that the device will be mapped to.
<code>name</code>	<code>string</code>	name must match the name of a persistentVolumeClaim in the pod

.spec.template.spec.initContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

.spec.template.spec.initContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`name` `mountPath`

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). Possible enum values: "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
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).
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.

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: <https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

object

.spec.template.spec.os

Description

PodOS defines the OS parameters of a pod.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration ↗ Clients should expect to handle additional values and treat unrecognized values in this field as os: null

.spec.template.spec.overhead

Description

Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info: <https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md>

Type

object

.spec.template.spec.readinessGates

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type

array

.spec.template.spec.readinessGates[]

Description

PodReadinessGate contains the reference to a pod condition

Type

object

Required

conditionType

Property	Type	Description
conditionType	string	ConditionType refers to a condition in the pod's condition list with matching type.

.spec.template.spec.resourceClaims

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.

Type

array

.spec.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.
resourceClaimTemplateName	string	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in pod.status.resourceClaimStatuses. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

.spec.template.spec.resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
claims	array	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.template.spec.resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.template.spec.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.template.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.template.spec.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.template.spec.schedulingGates`

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

`array`

`.spec.template.spec.schedulingGates[]`

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the scheduling gate. Each scheduling gate must have a unique name field.

`.spec.template.spec.securityContext`

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the 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. Possible enum values: "Always" indicates that volume's ownership and permissions should always be changed whenever volume is mounted inside a Pod. This the default behavior. "OnRootMismatch" indicates that volume's ownership and permissions will be changed only when permission and ownership of root directory does not match with expected permissions on the volume. This can help shorten the time it takes to change ownership and permissions of a volume.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
seLinuxChangePolicy	string	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.seLinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
supplementalGroups	array	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
supplementalGroupsPolicy	string	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows. Possible enum values: "Merge" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be merged with the primary user's groups as defined in the container image (in /etc/group).

Property	Type	Description
		"Strict" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be used instead of any groups defined in the container image.
sysctls	array	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

.spec.template.spec.securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

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. Possible enum values: "Localhost" indicates that a profile pre-loaded on the node should be used. "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. "Unconfined" indicates that no AppArmor profile should be enforced.

.spec.template.spec.securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.

Property	Type	Description
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

.spec.template.spec.securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	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.
<code>type</code>	<code>string</code>	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. Possible enum values: <code>"Localhost"</code> indicates a profile defined in a file on the node should be used. The file's location relative to /seccomp. <code>"RuntimeDefault"</code> represents the default container runtime seccomp profile. <code>"Unconfined"</code> indicates no seccomp profile is applied (A.K.A. unconfined).

.spec.template.spec.securityContext.supplementalGroups

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

`array`

.spec.template.spec.securityContext.supplementalGroups[]

Type

integer

.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

.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

.spec.template.spec.securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

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 <code>GMSACredentialSpecName</code> field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.template.spec.tolerations

Description

If specified, the pod's tolerations.

Type

array

.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. Possible enum values: "NoExecute" Evict any already-running pods that do not tolerate the taint. Currently enforced by NodeController. "NoSchedule" Do not allow new pods to schedule onto the node unless they tolerate the taint, but allow all pods submitted to Kubelet without going through the scheduler to start, and allow all already-running pods to continue running. Enforced by the scheduler. "PreferNoSchedule" Like TaintEffectNoSchedule, but the scheduler tries not to schedule new pods onto the node, rather than prohibiting new pods from scheduling onto the node entirely. Enforced by the scheduler.
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
		Possible enum values: "Equal" "Exists"
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
value	string	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.template.spec.topologySpreadConstraints

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

array

.spec.template.spec.topologySpreadConstraints[]

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

object

Required

maxSkew topologyKey whenUnsatisfiable

Property	Type	Description
labelSelector	object	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.
matchLabelKeys	array	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 <code>whenUnsatisfiable=DoNotSchedule</code>, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P - if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1). - if MaxSkew is 2, incoming pod can be scheduled onto any zone. When <code>whenUnsatisfiable=ScheduleAnyway</code>, it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.
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.
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. Possible enum values: <code>"Honor"</code> means use this scheduling directive when calculating pod topology spread skew. <code>"Ignore"</code> means ignore this scheduling directive when calculating pod topology spread skew.
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. Possible enum values: <code>"Honor"</code> means use this scheduling directive when calculating pod topology spread skew. <code>"Ignore"</code> means ignore this scheduling directive when calculating pod topology spread skew.

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	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.
<code>whenUnsatisfiable</code>	<code>string</code>	WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. - DoNotSchedule (default) tells the scheduler not to schedule it. - ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: zone1 zone2 zone3 P P P P P If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it <i>more</i> imbalanced. It's a required field. Possible enum values: <code>"DoNotSchedule"</code> instructs the scheduler not to schedule the pod when constraints are not satisfied. <code>"ScheduleAnyway"</code> instructs the scheduler to schedule the pod even if constraints are not satisfied.

`.spec.template.spec.topologySpreadConstraints[].labelSelector`

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
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

`array`

.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]

Type

string

.spec.template.spec.topologySpreadConstraints[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.template.spec.topologySpreadConstraints[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).

Type

array

`.spec.template.spec.topologySpreadConstraints[].matchLabelKeys[]`

Type

string

`.spec.template.spec.volumes`

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

`.spec.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
		Represents a Persistent Disk resource in AWS.
<code>awsElasticBlockStore</code>	<code>object</code>	An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.
<code>azureDisk</code>	<code>object</code>	AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.
<code>azureFile</code>	<code>object</code>	AzureFile represents an Azure File Service mount on the host and bind mount to the pod.
<code>cephfs</code>	<code>object</code>	Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.
<code>cinder</code>	<code>object</code>	Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership

Property	Type	Description
		management and SELinux relabeling.
<code>configMap</code>	<code>object</code>	Adapts a ConfigMap into a volume. The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.
<code>csi</code>	<code>object</code>	Represents a source location of a volume to mount, managed by an external CSI driver
<code>downwardAPI</code>	<code>object</code>	DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.
<code>emptyDir</code>	<code>object</code>	Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.
<code>ephemeral</code>	<code>object</code>	Represents an ephemeral volume that is handled by a normal storage driver.
<code>fc</code>	<code>object</code>	Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.
<code>flexVolume</code>	<code>object</code>	FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.
<code>flocker</code>	<code>object</code>	Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.
<code>gcePersistentDisk</code>	<code>object</code>	Represents a Persistent Disk resource in Google Compute Engine. A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.
<code>gitRepo</code>	<code>object</code>	Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling. 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.

Property	Type	Description
<code>glusterfs</code>	<code>object</code>	Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.
<code>hostPath</code>	<code>object</code>	Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.
<code>image</code>	<code>object</code>	ImageVolumeSource represents a image volume resource.
<code>iscsi</code>	<code>object</code>	Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.
<code>name</code>	<code>string</code>	name of the volume. Must be a DNS_LABEL and unique within the pod. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>nfs</code>	<code>object</code>	Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.
<code>persistentVolumeClaim</code>	<code>object</code>	PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).
<code>photonPersistentDisk</code>	<code>object</code>	Represents a Photon Controller persistent disk resource.
<code>portworxVolume</code>	<code>object</code>	PortworxVolumeSource represents a Portworx volume resource.
<code>projected</code>	<code>object</code>	Represents a projected volume source
<code>quobyte</code>	<code>object</code>	Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.
<code>rbd</code>	<code>object</code>	Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.
<code>scaleIO</code>	<code>object</code>	ScaleIOVolumeSource represents a persistent ScaleIO volume

Property	Type	Description
<code>secret</code>	<code>object</code>	Adapts a Secret into a volume. The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.
<code>storageos</code>	<code>object</code>	Represents a StorageOS persistent volume resource.
<code>vsphereVolume</code>	<code>object</code>	Represents a vSphere volume resource.

`.spec.template.spec.volumes[].awsElasticBlockStore`

Description

Represents a Persistent Disk resource in AWS. An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.

Type

`object`

Required

`volumeID`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
<code>partition</code>	<code>integer</code>	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).
<code>readOnly</code>	<code>boolean</code>	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
<code>volumeID</code>	<code>string</code>	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

`.spec.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
cacheMode	string	cacheMode is the Host Caching mode: None, Read Only, Read Write. Possible enum values: "None" "ReadOnly" "ReadWrite"
diskName	string	diskName is the Name of the data disk in the blob storage
diskURI	string	diskURI is the URI of data disk in the blob storage
fsType	string	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
kind	string	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared Possible enum values: "Dedicated" "Managed" "Shared"
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

```
.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
<code>readOnly</code>	<code>boolean</code>	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>secretName</code>	<code>string</code>	secretName is the name of secret that contains Azure Storage Account Name and Key
<code>shareName</code>	<code>string</code>	shareName is the azure share Name

.spec.template.spec.volumes[].cephfs

Description

Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.

Type

`object`

Required

`monitors`

Property	Type	Description
<code>monitors</code>	<code>array</code>	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>path</code>	<code>string</code>	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
<code>readOnly</code>	<code>boolean</code>	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 ↗
<code>secretFile</code>	<code>string</code>	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 ↗
<code>secretRef</code>	<code>object</code>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
<code>user</code>	<code>string</code>	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

.spec.template.spec.volumes[].cephfs.monitors

Description

monitors is Required: Monitors is a collection of Ceph monitors More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

array

.spec.template.spec.volumes[].cephfs.monitors[]

Type

string

.spec.template.spec.volumes[].cephfs.secretRef

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 ↗

.spec.template.spec.volumes[].cinder

Description

Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

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	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
volumeID	string	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

`.spec.template.spec.volumes[].cinder.secretRef`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.template.spec.volumes[].configMap`

Description

Adapts a ConfigMap into a volume. The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	optional specify whether the ConfigMap or its keys must be defined

`.spec.template.spec.volumes[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

.spec.template.spec.volumes[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

key path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.template.spec.volumes[].csi

Description

Represents a source location of a volume to mount, managed by an external CSI driver

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
<code>nodePublishSecretRef</code>	<code>object</code>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
<code>readOnly</code>	<code>boolean</code>	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
<code>volumeAttributes</code>	<code>object</code>	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

.spec.template.spec.volumes[].csi.nodePublishSecretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.template.spec.volumes[].csi.volumeAttributes

Description

volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

`object`

.spec.template.spec.volumes[].downwardAPI

Description

DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.

Type

`object`

Property	Type	Description
defaultMode	integer	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	Items is a list of downward API volume file

.spec.template.spec.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

array

.spec.template.spec.volumes[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
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	ResourceFieldSelector represents container resources (cpu, memory) and their output format

.spec.template.spec.volumes[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

.spec.template.spec.volumes[].downwardAPI.items[].resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number less than 10⁻³¹. When a Quantity is parsed from a string, it will remember the type of suffix it had, and will use that type when serializing. Before serializing, Quantity will be put in "canonical form". This means that Exponent and Suffix will be normalized to the following: - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. It is always a fixed-point representation of a number. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form when serialized. This format is intended to make it difficult to use these numbers without writing something like:
resource	string	Required: resource to select

.spec.template.spec.volumes[].emptyDir

Description

Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium.
sizeLimit	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and YAML. The serialization format is: The serialization format is:

Property	Type	Description
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> <integer> ::= <digit> <digit><digits> <digit><digits>.<digits> <digit><digits>.<digits><decimalSI> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.html) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number When a Quantity is parsed from a string, it will remember the type of suffix it had, and Before serializing, Quantity will be put in "canonical form". This means that Exponent/Prefix - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be emitted The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. Non-canonical values will still parse as long as they are well formed, but will be re-emitted in canonical form. This format is intended to make it difficult to use these numbers without writing some :

.spec.template.spec.volumes[].ephemeral

Description

Represents an ephemeral volume that is handled by a normal storage driver.

Type

object

Property	Type	Description
volumeClaimTemplate	object	PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate

Description

PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

Type

object

Required

spec

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

Description

PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.
dataSourceRef	object	TypedObjectReference contains enough information to let you locate the typed referenced object
resources	object	VolumeResourceRequirements describes the storage resource requirements for a volume.
selector	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.
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
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

Property	Type	Description
		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. Possible enum values: "Block" means the volume will not be formatted with a filesystem and will remain a raw block device. "Filesystem" means the volume will be or is formatted with a filesystem.
volumeName	string	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]

Type

string

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

Description

TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

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

Property	Type	Description
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef`

Description

TypedObjectReference contains enough information to let you locate the typed referenced object

Type

`object`

Required

`kind` `name`

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced
<code>namespace</code>	<code>string</code>	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources`

Description

VolumeResourceRequirements describes the storage resource requirements for a volume.

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Property	Type	Description
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]`

Type

string

`.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.template.spec.volumes[].fc

Description

Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
lun	integer	lun is Optional: FC target lun number
readOnly	boolean	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
targetWWNs	array	targetWWNs is Optional: FC target worldwide names (WWNs)
wwids	array	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

.spec.template.spec.volumes[].fc.targetWWNs

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

array

.spec.template.spec.volumes[].fc.targetWWNs[]

Type

string

.spec.template.spec.volumes[].fc.wwids

Description

wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

Type

array

.spec.template.spec.volumes[].fc.wwids[]

Type

string

.spec.template.spec.volumes[].flexVolume

Description

FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type

object

Required

driver

Property	Type	Description
driver	string	driver is the name of the driver to use for this volume.
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
options	object	options is Optional: this field holds extra command options if any.
readOnly	boolean	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

.spec.template.spec.volumes[].flexVolume.options

Description

options is Optional: this field holds extra command options if any.

Type

object

.spec.template.spec.volumes[].flexVolume.secretRef

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:

Property	Type	Description
https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗		

`.spec.template.spec.volumes[].flocker`

Description

Represents a Flocker volume mounted by the Flocker agent. One and only one of `datasetName` and `datasetUUID` should be set. Flocker volumes do not support ownership management or SELinux relabeling.

Type

object

Property	Type	Description
<code>datasetName</code>	string	<code>datasetName</code> is Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
<code>datasetUUID</code>	string	<code>datasetUUID</code> is the UUID of the dataset. This is unique identifier of a Flocker dataset

`.spec.template.spec.volumes[].gcePersistentDisk`

Description

Represents a Persistent Disk resource in Google Compute Engine. A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.

Type

object

Required

`pdName`

Property	Type	Description
<code>fsType</code>	string	<code>fsType</code> is filesystem type of the volume that you want to mount. 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 ↗
<code>partition</code>	integer	<code>partition</code> is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume <code>/dev/sda1</code> , you specify the partition as "1". Similarly, the volume partition for <code>/dev/sda</code> is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗
<code>pdName</code>	string	<code>pdName</code> is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗

`.spec.template.spec.volumes[].gitRepo`

Description

Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
<code>directory</code>	<code>string</code>	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
<code>repository</code>	<code>string</code>	repository is the URL
<code>revision</code>	<code>string</code>	revision is the commit hash for the specified revision.

`.spec.template.spec.volumes[].glusterfs`

Description

Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.

Type

`object`

Required

`endpoints` `path`

Property	Type	Description
<code>endpoints</code>	<code>string</code>	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗
<code>path</code>	<code>string</code>	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

`.spec.template.spec.volumes[].hostPath`

Description

Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath Possible enum values: <code>""</code> For backwards compatible, leave it empty if unset <code>"BlockDevice"</code> A block device must exist at the given path <code>"CharDevice"</code> A character device must exist at the given path <code>"Directory"</code> A directory must exist at the given path <code>"DirectoryOrCreate"</code> If nothing exists at the given path, an empty directory will be created there as needed with file mode 0755, having the same group and ownership with Kubelet. <code>"File"</code> A file must exist at the given path <code>"FileOrCreate"</code> If nothing exists at the given path, an empty file will be created there as needed with file mode 0644, having the same group and ownership with Kubelet. <code>"Socket"</code> A UNIX socket must exist at the given path

`.spec.template.spec.volumes[].image`

Description

ImageVolumeSource represents a image volume resource.

Type

`object`

Property	Type	Description
<code>pullPolicy</code>	<code>string</code>	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact.

Property	Type	Description
		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. Possible enum values: "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
reference	string	Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images ^ This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

.spec.template.spec.volumes[].iscsi

Description

Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.

Type

object

Required

targetPortal iqn lun

Property	Type	Description
chapAuthDiscovery	boolean	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
chapAuthSession	boolean	chapAuthSession defines whether support iSCSI Session CHAP authentication
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 : will be created for the connection.
iqn	string	iqn is the target iSCSI Qualified Name.

Property	Type	Description
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	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	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
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).

.spec.template.spec.volumes[].iscsi.portals

Description

portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

Type

array

.spec.template.spec.volumes[].iscsi.portals[]

Type

string

.spec.template.spec.volumes[].iscsi.secretRef

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

.spec.template.spec.volumes[].nfs

Description

Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.

Type

object

Required

server path

Property	Type	Description
path	string	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ^
readOnly	boolean	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ^
server	string	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ^

.spec.template.spec.volumes[].persistentVolumeClaim

Description

PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).

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.

.spec.template.spec.volumes[].photonPersistentDisk

Description

Represents a Photon Controller persistent disk resource.

Type

object

Required

pdID

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
pdID	string	pdID is the ID that identifies Photon Controller persistent disk

.spec.template.spec.volumes[].portworxVolume

Description

PortworxVolumeSource represents a Portworx volume resource.

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fSType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

.spec.template.spec.volumes[].projected

Description

Represents a projected volume source

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.

Property	Type	Description
<code>sources</code>	<code>array</code>	<code>sources</code> is the list of volume projections. Each entry in this list handles one source.

`.spec.template.spec.volumes[].projected.sources`

Description

`sources` is the list of volume projections. Each entry in this list handles one source.

Type

`array`

`.spec.template.spec.volumes[].projected.sources[]`

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

`object`

Property	Type	Description
<code>clusterTrustBundle</code>	<code>object</code>	<code>ClusterTrustBundleProjection</code> describes how to select a set of <code>ClusterTrustBundle</code> objects and project their contents into the pod filesystem.
<code>configMap</code>	<code>object</code>	Adapts a <code>ConfigMap</code> into a projected volume. The contents of the target <code>ConfigMap</code> 's <code>Data</code> field will be presented in a projected volume as files using the keys in the <code>Data</code> field as the file names, unless the <code>items</code> element is populated with specific mappings of keys to paths. Note that this is identical to a <code>configmap</code> volume source without the default mode.
<code>downwardAPI</code>	<code>object</code>	Represents downward API info for projecting into a projected volume. Note that this is identical to a <code>downwardAPI</code> volume source without the default mode.
<code>secret</code>	<code>object</code>	Adapts a secret into a projected volume. The contents of the target <code>Secret</code> 's <code>Data</code> field will be presented in a projected volume as files using the keys in the <code>Data</code> field as the file names. Note that this is identical to a <code>secret</code> volume source without the default mode.
<code>serviceAccountToken</code>	<code>object</code>	<code>ServiceAccountTokenProjection</code> represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

`.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle`

Description

ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.

Type

object

Required

path

Property	Type	Description
labelSelector	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.
name	string	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
optional	boolean	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
path	string	Relative path from the volume root to write the bundle.
signerName	string	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector

Description

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.template.spec.volumes[].projected.sources[].configMap`

Description

Adapts a ConfigMap into a projected volume. The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

`.spec.template.spec.volumes[].projected.sources[].configMap.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

`.spec.template.spec.volumes[].projected.sources[].configMap.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

key path

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.template.spec.volumes[].projected.sources[].downwardAPI

Description

Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

Type

object

Property	Type	Description
items	array	Items is a list of DownwardAPIVolume file

.spec.template.spec.volumes[].projected.sources[].downwardAPI.items

Description

Items is a list of DownwardAPIVolume file

Type

array

.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.

Property	Type	Description
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format

.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

fieldPath

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

resource

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	string number	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and

Property	Type	Description
		The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 1 ... 9 <digits> ::= <digit> <digit><digits> (International System of units; See: http://physics.nist.gov/cuu/Units/binary.htm) <decimalSI> ::= m "" k M G T P E (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.) <decimalExponent> ::= "e" <signedNumber> "E" <signedNumber> `` No matter which of the three exponent forms is used, no quantity may represent a number less than 10⁻³⁰. When a Quantity is parsed from a string, it will remember the type of suffix it had, and will use that type when serializing. For example, 1.5m will be serialized as "1500m". Before serializing, Quantity will be put in "canonical form". This means that Exponent notation will only be used when necessary to represent a number less than 10⁻³⁰. - No precision is lost - No fractional digits will be emitted - The exponent (or suffix) will be emitted when necessary to represent a number less than 10⁻³⁰. The sign will be omitted unless the number is negative. Examples: - 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi" Note that the quantity will NEVER be internally represented by a floating point number. All internal calculations are done on integers. Non-canonical values will still parse as long as they are well formed, but will be rounded to canonical form when serialized. This format is intended to make it difficult to use these numbers without writing scientific notation.
resource	string	Required: resource to select

.spec.template.spec.volumes[].projected.sources[].secret

Description

Adapts a secret into a projected volume. The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.

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 '..'.

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
<code>optional</code>	<code>boolean</code>	optional field specify whether the Secret or its key must be defined

`.spec.template.spec.volumes[].projected.sources[].secret.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

`array`

`.spec.template.spec.volumes[].projected.sources[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the key to project.
<code>mode</code>	<code>integer</code>	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.
<code>path</code>	<code>string</code>	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

`.spec.template.spec.volumes[].projected.sources[].serviceAccountToken`

Description

ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

Type

object

Required

path

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours.Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

.spec.template.spec.volumes[].quobyte

Description

Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

Type

object

Required

registry volume

Property	Type	Description
group	string	group to map volume access to Default is no group
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user

Property	Type	Description
volume	string	volume is a string that references an already created Quobyte volume by name.

.spec.template.spec.volumes[].rbd

Description

Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.

Type

object

Required

monitors image

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitors	array	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
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	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
user	string	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

.spec.template.spec.volumes[].rbd.monitors

Description

monitors is a collection of Ceph monitors. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

array

.spec.template.spec.volumes[].rbd.monitors[]

Type

string

.spec.template.spec.volumes[].rbd.secretRef

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

.spec.template.spec.volumes[].scaleIO

Description

ScaleIOVolumeSource represents a persistent ScaleIO volume

Type

object

Required

gateway system secretRef

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.

Property	Type	Description
<code>secretRef</code>	<code>object</code>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
<code>sslEnabled</code>	<code>boolean</code>	sslEnabled Flag enable/disable SSL communication with Gateway, default false
<code>storageMode</code>	<code>string</code>	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
<code>storagePool</code>	<code>string</code>	storagePool is the ScaleIO Storage Pool associated with the protection domain.
<code>system</code>	<code>string</code>	system is the name of the storage system as configured in ScaleIO.
<code>volumeName</code>	<code>string</code>	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

.spec.template.spec.volumes[].scaleIO.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.template.spec.volumes[].secret

Description

Adapts a Secret into a volume. The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.

Type

`object`

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON

Property	Type	Description
		requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
optional	boolean	optional field specify whether the Secret or its keys must be defined
secretName	string	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

.spec.template.spec.volumes[].secret.items

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

.spec.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.

Property	Type	Description
<code>path</code>	<code>string</code>	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.template.spec.volumes[].storageos

Description

Represents a StorageOS persistent volume resource.

Type

`object`

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
<code>readOnly</code>	<code>boolean</code>	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>secretRef</code>	<code>object</code>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
<code>volumeName</code>	<code>string</code>	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
<code>volumeNamespace</code>	<code>string</code>	volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

.spec.template.spec.volumes[].storageos.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.template.spec.volumes[].vsphereVolume

Description

Represents a vSphere volume resource.

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

.status

Description

DeploymentStatus is the most recently observed status of the Deployment.

Type

object

Property	Type	Description
availableReplicas	integer	Total number of available pods (ready for at least minReadySeconds) targeted by this deployment.
collisionCount	integer	Count of hash collisions for the Deployment. The Deployment controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ReplicaSet.
conditions	array	Represents the latest available observations of a deployment's current state.
observedGeneration	integer	The generation observed by the deployment controller.
readyReplicas	integer	readyReplicas is the number of pods targeted by this Deployment with a Ready Condition.

Property	Type	Description
<code>replicas</code>	<code>integer</code>	Total number of non-terminated pods targeted by this deployment (their labels match the selector).
<code>unavailableReplicas</code>	<code>integer</code>	Total number of unavailable pods targeted by this deployment. This is the total number of pods that are still required for the deployment to have 100% available capacity. They may either be pods that are running but not yet available or pods that still have not been created.
<code>updatedReplicas</code>	<code>integer</code>	Total number of non-terminated pods targeted by this deployment that have the desired template spec.

.status.conditions

Description

Represents the latest available observations of a deployment's current state.

Type

`array`

.status.conditions[]

Description

DeploymentCondition describes the state of a deployment at a certain point.

Type

`object`

Required

`type` `status`

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.
<code>lastUpdateTime</code>	<code>string</code>	Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.
<code>message</code>	<code>string</code>	A human readable message indicating details about the transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.

Property	Type	Description
type	string	Type of deployment condition.

API Endpoints

The following API endpoints are available:

- `/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/deployments`
 - `DELETE` : delete collection of Deployment
 - `GET` : list objects of kind Deployment
 - `POST` : create a new Deployment
- `/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/deployments/{name}`
 - `DELETE` : delete the specified Deployment
 - `GET` : read the specified Deployment
 - `PATCH` : partially update the specified Deployment
 - `PUT` : replace the specified Deployment
- `/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/deployments/{name}/status`
 - `GET` : read status of the specified Deployment
 - `PATCH` : partially update status of the specified Deployment
 - `PUT` : replace status of the specified Deployment

/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/deployments

HTTP method

DELETE

Description

delete collection of Deployment

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind Deployment

HTTP responses

HTTP code	Response body
200 - OK	DeploymentList schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new Deployment

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	Deployment schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	Deployment schema
201 - Created	Deployment schema
202 - Accepted	Deployment schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/deployments/{name}

HTTP method

DELETE

Description

delete the specified Deployment

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Deployment

HTTP responses

HTTP code	Response body
200 - OK	<code>Deployment</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Deployment

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>Deployment</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Deployment

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	Deployment schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	Deployment schema
201 - Created	Deployment schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/apis/apps/v1/namespaces/{namespace}/deployments/{name}/status

HTTP method

GET

Description

read status of the specified Deployment

HTTP responses

HTTP code	Response body
200 - OK	Deployment schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Deployment

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>Deployment</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Deployment

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>Deployment</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>Deployment</code> schema
201 - Created	<code>Deployment</code> schema
401 - Unauthorized	Empty