



PostgreSQL APIs

[Postgresqls](#)

[PostgresBackup](#)

[PostgresRe](#)



PostgreSQLs

Description

Postgresql defines a PostgreSQL cluster managed by the Zalando Postgres Operator. It allows configuring various aspects of PostgreSQL including replication, backups, resource allocation, and more. Postgresql defines the desired state and status of a PostgreSQL cluster managed by the Zalando Postgres Operator.

Type

`object`

Required

`spec`

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

Property	Type	Description
		endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
metadata	ObjectMeta ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	Spec defines the desired state of the PostgreSQL cluster including configuration, resources, replication settings, and more.
status	object	Status defines the observed state of the PostgreSQL cluster including current status, patroni state, and any error messages.

.spec

Description

Spec defines the desired state of the PostgreSQL cluster including configuration, resources, replication settings, and more.

Type

object

Required

numberOfInstances

postgresql

teamId

Property	Type	Description
<code>additionalVolumes</code>	<code>array</code>	AdditionalVolumes defines additional volumes for the pod
<code>allowedSourceRanges</code>	<code>array</code>	AllowedSourceRanges specifies IP ranges allowed to access load balancers
<code>backup</code>	<code>object</code>	Backup defines backup configuration
<code>clone</code>	<code>object</code>	Clone defines configuration for cloning from another cluster
<code>clusterReplication</code>	<code>object</code>	ClusterReplication defines cross-cluster replication
<code>connectionPooler</code>	<code>object</code>	ConnectionPooler configuration for connection pooling
<code>databases</code>	<code>object</code>	Databases defines databases to be created in the cluster

Property	Type	Description
<code>dockerImage</code>	<code>string</code>	DockerImage specifies the container image to use for PostgreSQL instances
<code>enableConnectionPooler</code>	<code>boolean</code>	EnableConnectionPooler enables connection pooling for the primary instance
<code>enableExporter</code>	<code>boolean</code>	EnableExporter enables Prometheus exporter
<code>enableLogicalBackup</code>	<code>boolean</code>	EnableLogicalBackup enables logical backups for the cluster
<code>enableMasterLoadBalancer</code>	<code>boolean</code>	EnableMasterLoadBalancer enables load balancer for master instance
<code>enableMasterPoolerLoadBalancer</code>	<code>boolean</code>	EnableMasterPoolerLoadBalancer enables load balancer for master pooler
<code>enablePgpool2</code>	<code>boolean</code>	EnablePgpool2 enables Pgpool-II connection pooling

Property	Type	Description
<code>enableReadinessProbe</code>	<code>boolean</code>	EnableReadinessProbe enables readiness probes for containers
<code>enableReplicaConnectionPooler</code>	<code>boolean</code>	EnableReplicaConnectionPooler enables connection pooling for replica instances
<code>enableReplicaLoadBalancer</code>	<code>boolean</code>	EnableReplicaLoadBalancer enables load balancer for replica instances
<code>enableReplicaPoolerLoadBalancer</code>	<code>boolean</code>	EnableReplicaPoolerLoadBalancer enables load balancer for replica pooler
<code>enableShmVolume</code>	<code>boolean</code>	ShmVolume enables shared memory volume for the pod
<code>env</code>	<code>array</code>	Env defines environment variables for containers
<code>exporter</code>	<code>object</code>	Exporter defines Prometheus exporter configuration

Property	Type	Description
<code>init_containers</code>	<code>array</code>	InitContainersOld is a deprecated field for init containers
<code>initContainers</code>	<code>array</code>	InitContainers defines initialization containers for the pod
<code>injectPass</code>	<code>boolean</code>	InjectPass enables password injection for users
<code>ipFamilyPrefer</code>	<code>string</code>	IPFamilyPrefer specifies preferred IP family for services
<code>logicalBackupSchedule</code>	<code>string</code>	LogicalBackupSchedule defines the cron schedule for logical backups
<code>maintenanceWindows</code>	<code>array</code>	MaintenanceWindows defines time windows for maintenance operations
<code>masterServiceAnnotations</code>	<code>object</code>	MasterServiceAnnotations defines annotations for master service
<code>nodeAffinity</code>	<code>object</code>	NodeAffinity defines node affinity rules for pod scheduling

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	NodeSelector defines node labels for pod scheduling
<code>numberOfInstances</code>	<code>integer</code>	NumberOfInstances specifies the number of PostgreSQL replicas
<code>patroni</code>	<code>object</code>	Patroni configuration for cluster management
<code>pgpool2Settings</code>	<code>object</code>	Pgpool2Settings defines Pgpool-II configuration
<code>pod_priority_class_name</code>	<code>string</code>	PodPriorityClassNameOld is a deprecated field for priority class
<code>podAnnotations</code>	<code>object</code>	PodAnnotations defines annotations to add to pods
<code>podPriorityClassName</code>	<code>string</code>	PodPriorityClassName specifies the priority class for pods
<code>postgresql</code>	<code>object</code>	PostgreSQL version and configuration parameters including: - PostgreSQL major version (e.g. "14") - Configuration parameters

Property	Type	Description
		(postgresql.conf) - Configuration parameters (postgresql.conf) PostgreSQL version and configuration parameters
<code>preparedDatabases</code>	<code>object</code>	PreparedDatabases defines databases with pre-configured schemas and roles
<code>repairOption</code>	<code>object</code>	RepairOption defines options for cluster repair
<code>replicaLoadBalancer</code>	<code>boolean</code>	ReplicaLoadBalancer is a deprecated field for replica load balancer
<code>replicaServiceAnnotations</code>	<code>object</code>	ReplicaServiceAnnotations defines annotations for replica service
<code>resources</code>	<code>object</code>	Resource requests and limits for PostgreSQL containers
<code>schedulerName</code>	<code>string</code>	SchedulerName specifies the Kubernetes scheduler to use

Property	Type	Description
<code>serviceAnnotations</code>	<code>object</code>	ServiceAnnotations defines annotations to add to services
<code>serviceTemplates</code>	<code>object</code>	ServiceTemplates defines custom service templates
<code>sidecars</code>	<code>array</code>	Sidecars defines additional containers to run in the pod
<code>spiloAllowPrivilegeEscalation</code>	<code>boolean</code>	SpiloAllowPrivilegeEscalation controls privilege escalation for Spilo
<code>spiloFSGroup</code>	<code>integer</code>	SpiloFSGroup specifies the filesystem group ID for Spilo container
<code>spiloPrivileged</code>	<code>boolean</code>	SpiloPrivileged enables privileged mode for Spilo container
<code>spiloReadOnlyRootFilesystem</code>	<code>boolean</code>	SpiloReadOnlyRootFilesystem enables read-only root filesystem for Spilo

Property	Type	Description
<code>spiloRunAsGroup</code>	<code>integer</code>	SpiloRunAsGroup specifies the group ID to run Spilo container as
<code>spiloRunAsUser</code>	<code>integer</code>	SpiloRunAsUser specifies the user ID to run Spilo container as
<code>standby</code>	<code>object</code>	StandbyCluster defines configuration for standby clusters
<code>streams</code>	<code>array</code>	Streams defines configuration for streaming data
<code>teamId</code>	<code>string</code>	TeamID is the identifier of the team owning the cluster
<code>tls</code>	<code>object</code>	TLS defines TLS configuration for the cluster
<code>tolerations</code>	<code>array</code>	Tolerations defines pod tolerations for node taints
<code>upgradeOption</code>	<code>object</code>	UpgradeOption defines upgrade options

Property	Type	Description
<code>useLoadBalancer</code>	<code>boolean</code>	UseLoadBalancer is a deprecated field for enabling load balancer
<code>users</code>	<code>object</code>	Users defines additional PostgreSQL users and their roles
<code>usersCustomizedPasswd</code>	<code>object</code>	UsersCustomizedPasswd stores custom passwords for users
<code>usersWithInPlaceSecretRotation</code>	<code>array</code>	UsersWithInPlaceSecretRotation defines users requiring in-place secret rotation
<code>usersWithSecretRotation</code>	<code>array</code>	UsersWithSecretRotation defines users requiring secret rotation
<code>volume</code>	<code>object</code>	Volume configuration for PostgreSQL data storage

`.spec.additionalVolumes`

Description

AdditionalVolumes defines additional volumes for the pod

Type

`array`

`.spec.additionalVolumes[]`

Description

AdditionalVolume defines additional volumes that can be mounted to PostgreSQL pods

Type

object

Required

mountPath

name

targetContainers

volumeSource

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path where the volume will be mounted in the container
<code>name</code>	<code>string</code>	Name of the volume, must be unique within the pod
<code>subPath</code>	<code>string</code>	SubPath within the volume to mount (optional)
<code>targetContainers</code>	<code>array</code>	List of container names that should mount this volume
<code>volumeSource</code>	<code>object</code>	Kubernetes VolumeSource defining the volume type and configuration

`.spec.additionalVolumes[].targetContainers`

Description

List of container names that should mount this volume

Type

array

`.spec.additionalVolumes[].targetContainers[]`

Type

string

`.spec.additionalVolumes[].volumeSource`

Description

Kubernetes VolumeSource defining the volume type and configuration

Type

object

Property	Type	Description
<code>awsElasticBlockStore</code>	object	awsElasticBlockStore represents an AWS Disk resource a kubelet's host machine and then exposed to the pod. https://kubernetes.io/docs/concepts/storage/volumes#aws-elastic-block-store
<code>azureDisk</code>	object	azureDisk represents an Azure Data Disk mount on the host and mounted on the pod.
<code>azureFile</code>	object	azureFile represents an Azure File Service mount on the host and mounted on the pod.
<code>cephfs</code>	object	cephFS represents a Ceph FS mount on the host that supports read/write operations.

Property	Type	Description
cinder	object	cinder represents a cinder volume attached and mount machine. More info: https://examples.k8s.io/mysql-cind 
configMap	object	configMap represents a configMap that should populate
csi	object	csi (Container Storage Interface) represents ephemera handled by certain external CSI drivers (Beta feature).
downwardAPI	object	downwardAPI represents downward API about the pod populate this volume
emptyDir	object	emptyDir represents a temporary directory that shares More info: https://kubernetes.io/docs/concepts/storage/ 
ephemeral	object	ephemeral represents a volume that is handled by a cl The volume's lifecycle is tied to the pod that defines it - before the pod starts, and deleted when the pod is rem the volume is only needed while the pod runs, b) featur volumes like restoring from snapshot or capacity trackin the storage driver is specified through a storage class, driver supports dynamic volume provisioning through a PersistentVolumeClaim (see EphemeralVolumeSource information on the connection between this volume typ PersistentVolumeClaim). Use PersistentVolumeClaim c vendor-specific APIs for volumes that persist for longer

Property	Type	Description
		an individual pod. Use CSI for light-weight local ephem CSI driver is meant to be used that way - see the docur driver for more information. A pod can use both types o volumes and persistent volumes at the same time.
fc	object	fc represents a Fibre Channel resource that is attached machine and then exposed to the pod.
flexVolume	object	flexVolume represents a generic volume resource that i provisioned/attached using an exec based plugin.
flocker	object	flocker represents a Flocker volume attached to a kube This depends on the Flocker control service being runn
gcePersistentDisk	object	gcePersistentDisk represents a GCE Disk resource tha kubelet's host machine and then exposed to the pod. M https://kubernetes.io/docs/concepts/storage/volumes#g
gitRepo	object	gitRepo represents a git repository at a particular revisi GitRepo is deprecated. To provision a container with a EmptyDir into an InitContainer that clones the repo usir the EmptyDir into the Pod's container.
glusterfs	object	glusterfs represents a Glusterfs mount on the host that lifetime. More info: https://examples.k8s.io/volumes/glu

Property	Type	Description
hostPath	object	hostPath represents a pre-existing file or directory on the host that is directly exposed to the container. This is general purpose and can be used by general agents or other privileged things that are allowed to see the host. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath TODO(jonesdl) We need to restrict who can use hostPath and who can/can not mount host directories as read/write
iscsi	object	iscsi represents an iSCSI Disk resource that is attached to the host machine and then exposed to the pod. More info: https://examples.k8s.io/volumes/iscsi/README.md
nfs	object	nfs represents an NFS mount on the host that shares a file system with info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
persistentVolumeClaim	object	persistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
photonPersistentDisk	object	photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine
portworxVolume	object	portworxVolume represents a portworx volume attached and mounted on kubelets host machine

Property	Type	Description
<code>projected</code>	<code>object</code>	projected items for all in one resources secrets, config downward API
<code>quobyte</code>	<code>object</code>	quobyte represents a Quobyte mount on the host that s lifetime
<code>rbd</code>	<code>object</code>	rbd represents a Rados Block Device mount on the hos pod's lifetime. More info: https://examples.k8s.io/volumes/rbd/README.md ↗
<code>scaleIO</code>	<code>object</code>	scaleIO represents a ScaleIO persistent volume attach Kubernetes nodes.
<code>secret</code>	<code>object</code>	secret represents a secret that should populate this vol https://kubernetes.io/docs/concepts/storage/volumes#s
<code>storageos</code>	<code>object</code>	storageOS represents a StorageOS volume attached a Kubernetes nodes.
<code>vsphereVolume</code>	<code>object</code>	vsphereVolume represents a vSphere volume attached kubelets host machine

.spec.additionalVolumes[].volumeSource.awsElasticBlock Store

Description

awsElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. 1 Ensure that the filesystem type is supported by the host operating sy Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspe More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblo ↗ TODO: how do we prevent errors in the filesystem from comprom the machine
partition	integer	partition is the partition in the volume that you want to mount. If omit the default is to mount by volume name. Examples: For volume /dev you specify the partition as "1". Similarly, the volume partition for /de is "0" (or you can leave the property empty).
readOnly	boolean	readOnly value true will force the readOnly setting in VolumeMounts info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblo ↗
volumeID	string	volumeID is unique ID of the persistent disk resource in AWS (Amaz EBS volume). More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore 

`.spec.additionalVolumes[].volumeSource.azureDisk`

Description

azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

object

Required

diskName

diskURI

Property	Type	Description
<code>cachedMode</code>	<code>string</code>	cachedMode is the Host Caching mode: None, Read Only, Read Write.
<code>diskName</code>	<code>string</code>	diskName is the Name of the data disk in the blob storage
<code>diskURI</code>	<code>string</code>	diskURI is the URI of data disk in the blob storage
<code>fsType</code>	<code>string</code>	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

Property	Type	Description
<code>kind</code>	<code>string</code>	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared
<code>readOnly</code>	<code>boolean</code>	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

`.spec.additionalVolumes[].volumeSource.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

Property	Type	Description
<code>shareName</code>	<code>string</code>	shareName is the azure share Name

`.spec.additionalVolumes[].volumeSource.cephfs`

Description

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

Type

`object`

Required

`monitors`

Property	Type	Description
<code>monitors</code>	<code>array</code>	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 ↗

Property	Type	Description
<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>	secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
<code>user</code>	<code>string</code>	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.cephfs.monitors`

`[]`

Type

`string`

`.spec.additionalVolumes[].volumeSource.cephfs.secretRef`

Description

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

Type

object

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

`.spec.additionalVolumes[].volumeSource.cinder`

Description

cinder represents a cinder volume attached and mounted on kubelets host machine. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if

Property	Type	Description
		unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> . More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is optional: points to a secret object containing parameters used to connect to OpenStack.
<code>volumeID</code>	<code>string</code>	<code>volumeID</code> used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

`.spec.additionalVolumes[].volumeSource.cinder.secretRef`

Description

`secretRef` is optional: points to a secret object containing parameters used to connect to OpenStack.

Type

`object`

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

.spec.additionalVolumes[].volumeSource.configMap

Description

configMap represents a configMap that should populate this volume

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

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

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.csi

Description

csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Type

object

Required

driver

Property	Type	Description
<code>driver</code>	<code>string</code>	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
<code>fsType</code>	<code>string</code>	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
<code>nodePublishSecretRef</code>	<code>object</code>	nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.
<code>readOnly</code>	<code>boolean</code>	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
<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.additionalVolumes[].volumeSource.csi.nodePublishSecretRef`

Description

nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

object

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

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.downwardAPI`

Description

downwardAPI represents downward API about the pod that should populate this volume

Type

object

Property	Type	Description
defaultMode	integer	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	Items is a list of downward API volume file

.spec.additionalVolumes[].volumeSource.downwardAPI.items**Description**

Items is a list of downward API volume file

Type

array

.spec.additionalVolumes[].volumeSource.downwardAPI.items[]**Description**

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

path

Property	Type	Description
fieldRef	object	Required: Selects a field of the pod: only annotations, labels, name and namespace are supported.
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

`.spec.additionalVolumes[].volumeSource.downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name and namespace are supported.

Type

`object`

Required

`fieldPath`

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

`.spec.additionalVolumes[].volumeSource.downwardAPI.items[].resourceFieldRef`

Description

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

Type

`object`

Required

`resource`

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

`.spec.additionalVolumes[].volumeSource.emptyDir`

Description

`emptyDir` represents a temporary directory that shares a pod's lifetime. More info: <https://kubernetes.io/docs/concepts/storage/volumes#emptydir>

Type

`object`

Property	Type	Description
<code>medium</code>	<code>string</code>	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
<code>sizeLimit</code>	<code>int</code>	<code>sizeLimit</code> is the total amount of local storage required for this <code>EmptyDir</code> volume. The size limit is also applicable for memory

Property	Type	Description
		medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.additionalVolumes[].volumeSource.ephemeral

Description

ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual pod. Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.

Type

object

Property	Type	Description
<code>volumeClaimTemplate</code>	object	Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted

Property	Type	Description
		together with the pod. The name of the PVC will be <code><pod name>-<volume name></code> where <code><volume name></code> is the name from the <code>PodSpec.Volumes</code> array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will <i>not</i> be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to be updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

`.spec.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate`

Description

Will be used to create a stand-alone PVC to provision the volume. The pod in which this `EphemeralVolumeSource` is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be ``<pod name>-<volume name>`` where ``<volume name>`` is the name from the ``PodSpec.Volumes`` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will **not** be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to be updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a

broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

Type

object

Required

spec

Property	Type	Description
metadata	ObjectMeta ↗	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.
spec	object	The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

.spec.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate.spec

Description

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such both fields are non-empty, they must have the same value. For backwards compatibility, when namespace

Property	Type	Description
		isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef. While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. While dataSource only allows local objects, dataSourceRef allows objects in any namespaces (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled, users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

Property	Type	Description
<code>selector</code>	<code>object</code>	selector is a label query over volumes to consider binding.
<code>storageClassName</code>	<code>string</code>	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1 ↗
<code>volumeAttributesClassName</code>	<code>string</code>	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified the CSI driver will create or update the volume with 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 VolumeAttributesClass will be applied to the claim it's not allowed to reset this field to empty string once it's set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be used by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#volumeattributesclass ↗ (Alpha) Using this field requires the VolumeAttributesClass feature gate to be enabled.

Property	Type	Description
<code>volumeMode</code>	<code>string</code>	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	volumeName is the binding reference to the PersistentVolume backing this claim.

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate.spec.accessModes[]`

Type

`string`

`.spec.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate.spec.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If

the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced

.spec.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate.spec.dataSourceRef

Description

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as

such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.

(Alpha) Using the namespace field of `dataSourceRef` requires the `CrossNamespaceVolumeDataSource` feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a

Property	Type	Description
		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.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate.spec.resources`

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

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

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

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate.spec.selector`

Description

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

Type

object

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

`.spec.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate.spec.selector.matchExpressions`

Description

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

Type

array

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values

s[]**Type**

string

.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.fc**Description**

fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Type

object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. TODO: how do we prevent errors in the filesystem from compromising the machine

Property	Type	Description
<code>lun</code>	<code>integer</code>	lun is Optional: FC target lun number
<code>readOnly</code>	<code>boolean</code>	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>targetWWNs</code>	<code>array</code>	targetWWNs is Optional: FC target worldwide names (WWNs)
<code>wwids</code>	<code>array</code>	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

`.spec.additionalVolumes[].volumeSource.fc.targetWWNs`

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

`array`

`.spec.additionalVolumes[].volumeSource.fc.targetWWNs[]`

Type

`string`

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.fc.wwids[]`

Type

string

`.spec.additionalVolumes[].volumeSource.flexVolume`

Description

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

Type

object

Required

driver

Property	Type	Description
<code>driver</code>	<code>string</code>	driver is the name of the driver to use for this volume.
<code>fsType</code>	<code>string</code>	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.

Property	Type	Description
<code>options</code>	<code>object</code>	<code>options</code> is Optional: this field holds extra command options if any.
<code>readOnly</code>	<code>boolean</code>	<code>readOnly</code> is Optional: defaults to false (read/write). <code>ReadOnly</code> here will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> .
<code>secretRef</code>	<code>object</code>	<code>secretRef</code> is Optional: <code>secretRef</code> is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

`.spec.additionalVolumes[].volumeSource.flexVolume.options`

Description

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

Type

`object`

`.spec.additionalVolumes[].volumeSource.flexVolume.secretRef`

Description

`secretRef` is Optional: `secretRef` is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified.

If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Type

object

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

.spec.additionalVolumes[].volumeSource.flocker

Description

flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Type

object

Property	Type	Description
datasetName	string	datasetName is Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
datasetUUID	string	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

.spec.additionalVolumes[].volumeSource.gcePersistentDisk

Description

gcePersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

Type

object

Required

pdName

Property	Type	Description
<code>fsType</code>	<code>string</code>	fsType is filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗ TODO: how do we prevent errors in the filesystem from compromising the machine
<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). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗

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

`.spec.additionalVolumes[].volumeSource.gitRepo`

Description

gitRepo represents a git repository at a particular revision. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

object

Required

repository

Property	Type	Description
directory	string	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.

Property	Type	Description
<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.additionalVolumes[].volumeSource.glusterfs`

Description

glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/glusterfs/README.md>

Type

`object`

Required

`endpoints`

`path`

Property	Type	Description
<code>endpoints</code>	<code>string</code>	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
<code>path</code>	<code>string</code>	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

Property	Type	Description
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗

`.spec.additionalVolumes[].volumeSource.hostPath`

Description

hostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath> --- TODO(jonesdl) We need to restrict who can use host directory mounts and who can/can not mount host directories as read/write.

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗

.spec.additionalVolumes[].volumeSource.iscsi

Description

iscsi represents an iSCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://examples.k8s.io/volumes/iscsi/README.md>

Type

object

Required

iqn lun targetPortal

Property	Type	Description
chapAuthDiscovery	boolean	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
chapAuthSession	boolean	chapAuthSession defines whether support iSCSI Session CHAP authentication
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 ⬆️ TODO: how do we prevent errors in the filesystem from compromising the machine

Property	Type	Description
<code>initiatorName</code>	<code>string</code>	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
<code>iqn</code>	<code>string</code>	iqn is the target iSCSI Qualified Name.
<code>iscsiInterface</code>	<code>string</code>	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
<code>lun</code>	<code>integer</code>	lun represents iSCSI Target Lun number.
<code>portals</code>	<code>array</code>	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
<code>secretRef</code>	<code>object</code>	secretRef is the CHAP Secret for iSCSI target and initiator authentication

Property	Type	Description
<code>targetPortal</code>	<code>string</code>	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.iscsi.portals[]`

Type

`string`

`.spec.additionalVolumes[].volumeSource.iscsi.secretRef`

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-

Property	Type	Description
		objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?

.spec.additionalVolumes[].volumeSource.nfs

Description

nfs represents an NFS mount on the host that shares a pod's lifetime More info:
<https://kubernetes.io/docs/concepts/storage/volumes#nfs>

Type

object

Required

path

server

Property	Type	Description
path	string	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗
readOnly	boolean	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗
server	string	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

`.spec.additionalVolumes[].volumeSource.persistentVolumeClaim`

Description

`persistentVolumeClaimVolumeSource` represents a reference to a `PersistentVolumeClaim` in the same namespace. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

object

Required

`claimName`

Property	Type	Description
<code>claimName</code>	string	<code>claimName</code> is the name of a <code>PersistentVolumeClaim</code> in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗
<code>readOnly</code>	boolean	<code>readOnly</code> Will force the <code>ReadOnly</code> setting in <code>VolumeMounts</code> . Default false.

`.spec.additionalVolumes[].volumeSource.photonPersistentDisk`

Description

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

Type

object

Required

pdID

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
pdID	string	pdID is the ID that identifies Photon Controller persistent disk

.spec.additionalVolumes[].volumeSource.portworxVolume

Description

portworxVolume represents a portworx volume attached and mounted on kubelets host machine

Type

object

Required

volumeID

Property	Type	Description
fsType	string	fSType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Property	Type	Description
<code>volumeID</code>	<code>string</code>	volumeID uniquely identifies a Portworx volume

`.spec.additionalVolumes[].volumeSource.projected`

Description

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

Type

`object`

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	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.
<code>sources</code>	<code>array</code>	sources is the list of volume projections

`.spec.additionalVolumes[].volumeSource.projected.sources`

Description

sources is the list of volume projections

Type

array

.spec.additionalVolumes[].volumeSource.projected.sources[]

Description

Projection that may be projected along with other supported volume types

Type

object

Property	Type	Description
clusterTrustBundle	object	ClusterTrustBundle allows a pod to access the .spec.trustBundle field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.
configMap	object	configMap information about the configMap data to project

Property	Type	Description
<code>downwardAPI</code>	<code>object</code>	downwardAPI information about the downwardAPI data to project
<code>secret</code>	<code>object</code>	secret information about the secret data to project
<code>serviceAccountToken</code>	<code>object</code>	serviceAccountToken is information about the serviceAccountToken data to project

`.spec.additionalVolumes[].volumeSource.projected.sources[].clusterTrustBundle`

Description

ClusterTrustBundle allows a pod to access the `.spec.trustBundle` field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.`

Type

`object`

Required

`path`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".
<code>name</code>	<code>string</code>	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
<code>optional</code>	<code>boolean</code>	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.
<code>path</code>	<code>string</code>	Relative path from the volume root to write the bundle.
<code>signerName</code>	<code>string</code>	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

`.spec.additionalVolumes[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector`

Description

Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Type

object

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

.spec.additionalVolumes[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector.matchExpressions

Description

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

Type

array

.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.projected.sources[].configMap`

Description

configMap information about the configMap data to project

Type

`object`

Property	Type	Description
<code>items</code>	<code>array</code>	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is

Property	Type	Description
		not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.projected.sources[].downwardAPI`

Description

downwardAPI information about the downwardAPI data to project

Type

object

Property	Type	Description
<code>items</code>	<code>array</code>	Items is a list of DownwardAPIVolume file

`.spec.additionalVolumes[].volumeSource.projected.sources[].downwardAPI.items`

Description

Items is a list of DownwardAPIVolume file

Type

`array`

`.spec.additionalVolumes[].volumeSource.projected.sources[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

`object`

Required

`path`

Property	Type	Description
<code>fieldRef</code>	<code>object</code>	Required: Selects a field of the pod: only annotations, labels, name and namespace are supported.
<code>mode</code>	<code>integer</code>	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or

Property	Type	Description
		a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
<code>path</code>	<code>string</code>	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

`.spec.additionalVolumes[].volumeSource.projected.sources[].downwardAPI.items[].fieldRef`

Description

Required: Selects a field of the pod: only annotations, labels, name and namespace are supported.

Type

`object`

Required

`fieldPath`

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

`.spec.additionalVolumes[].volumeSource.projected.sources[].downwardAPI.items[].resourceFieldRef`

Description

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

Type

`object`

Required

`resource`

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

`.spec.additionalVolumes[].volumeSource.projected.sources[].secret`

Description

secret information about the secret data to project

Type

object

Property	Type	Description
<code>items</code>	<code>array</code>	items if unspecified, each key-value pair in the Data field of the referenced 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 '..'.
<code>name</code>	<code>string</code>	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names [↗] TODO: Add other useful fields. apiVersion, kind, uid?
<code>optional</code>	<code>boolean</code>	optional field specify whether the Secret or its key must be defined

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.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.

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.additionalVolumes[].volumeSource.projected.sources[].serviceAccountToken`

Description

serviceAccountToken is information about the serviceAccountToken data to project

Type

`object`

Required

`path`

Property	Type	Description
<code>audience</code>	<code>string</code>	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.
<code>expirationSeconds</code>	<code>integer</code>	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

Property	Type	Description
		token is older than 24 hours.Defaults to 1 hour and must be at least 10 minutes.
<code>path</code>	<code>string</code>	path is the path relative to the mount point of the file to project the token into.

`.spec.additionalVolumes[].volumeSource.quobyte`

Description

quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Type

`object`

Required

`registry`

`volume`

Property	Type	Description
<code>group</code>	<code>string</code>	group to map volume access to Default is no group
<code>readOnly</code>	<code>boolean</code>	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
<code>registry</code>	<code>string</code>	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

Property	Type	Description
<code>tenant</code>	<code>string</code>	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
<code>user</code>	<code>string</code>	user to map volume access to Defaults to serviceaccount user
<code>volume</code>	<code>string</code>	volume is a string that references an already created Quobyte volume by name.

`.spec.additionalVolumes[].volumeSource.rbd`

Description

rbd represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/rbd/README.md>

Type

`object`

Required

`image`

`monitors`

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd ↗ TODO: how do we prevent errors in the filesystem from compromising the machine
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:

Property	Type	Description
		https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
<code>secretRef</code>	<code>object</code>	secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
<code>user</code>	<code>string</code>	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.rbd.monitors[]`

Type

`string`

`.spec.additionalVolumes[].volumeSource.rbd.secretRef`

Description

secretRef is name of the authentication secret for RBDUser. If provided overrides keyring.
Default is nil. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

Type

object

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

.spec.additionalVolumes[].volumeSource.scaleIO

Description

scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Type

object

Required

gateway

secretRef

system

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
gateway	string	gateway is the host address of the ScaleIO API Gateway.

Property	Type	Description
<code>protectionDomain</code>	<code>string</code>	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
<code>readOnly</code>	<code>boolean</code>	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
<code>secretRef</code>	<code>object</code>	secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.
<code>sslEnabled</code>	<code>boolean</code>	sslEnabled Flag enable/disable SSL communication with Gateway, default false
<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

Property	Type	Description
		volume source.

`.spec.additionalVolumes[].volumeSource.scaleIO.secretRef`

Description

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

object

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

`.spec.additionalVolumes[].volumeSource.secret`

Description

secret represents a secret that should populate this volume. More info:
<https://kubernetes.io/docs/concepts/storage/volumes#secret>

Type

object

Property	Type	Description
<code>defaultMode</code>	<code>integer</code>	<code>defaultMode</code> 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 <code>fsGroup</code>, and the result can be other mode bits set.
<code>items</code>	<code>array</code>	<code>items</code> If unspecified, each key-value pair in the <code>Data</code> field of the referenced <code>Secret</code> 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 <code>Secret</code>, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the <code>..'</code> path or start with <code>..'</code>.
<code>optional</code>	<code>boolean</code>	<code>optional</code> field specify whether the <code>Secret</code> or its keys must be defined
<code>secretName</code>	<code>string</code>	<code>secretName</code> is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

`.spec.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.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.additionalVolumes[].volumeSource.storageos`

Description

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

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>	secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.
<code>volumeName</code>	<code>string</code>	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique

Property	Type	Description
		within a namespace.
volumeNamespace	string	volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

.spec.additionalVolumes[].volumeSource.storageos.secretRef

Description

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

object

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

.spec.additionalVolumes[].volumeSource.vsphereVolume

Description

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

Type

object

Required

volumePath

Property	Type	Description
fsType	string	fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
storagePolicyID	string	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
storagePolicyName	string	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
volumePath	string	volumePath is the path that identifies vSphere volume vmdk

.spec.allowedSourceRanges

Description

AllowedSourceRanges specifies IP ranges allowed to access load balancers

Type

array

.spec.allowedSourceRanges[]**Type**

string

.spec.backup**Description**

Backup defines backup configuration

Type

object

Required

storage

Property	Type	Description
resources	object	Resources describes requests and limits for the Backup sidecar resources.
retainDay	integer	RetainDay specifies how many days to retain backup files before deletion
schedule	string	Schedule defines the cron schedule for automated backups (e.g. "0 0 * * *" for daily at midnight)

Property	Type	Description
<code>storage</code>	<code>object</code>	storage defines the configuration for backup storage including the storage name, namespace, and bucket location.

`.spec.backup.resources`

Description

Resources describes requests and limits for the Backup sidecar resources.

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	ResourceLimits defines the maximum resources allowed for containers
<code>requests</code>	<code>object</code>	ResourceRequests defines the minimum resources required for containers

`.spec.backup.resources.limits`

Description

ResourceLimits defines the maximum resources allowed for containers

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

.spec.backup.resources.requests

Description

ResourceRequests defines the minimum resources required for containers

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

.spec.backup.storage

Description

storage defines the configuration for backup storage including the storage name, namespace, and bucket location.

Type

`object`

Required

bucket

name

namespace

Property	Type	Description
bucket	string	S3 bucket name
name	string	Name of the storage configuration to use for backups
namespace	string	Namespace where the storage configuration is defined

.spec.clone

Description

Clone defines configuration for cloning from another cluster

Type

object

Property	Type	Description
cluster	string	ClusterName specifies the name of the source cluster to clone from
s3_access_key_id	string	S3AccessKeyId specifies the access key ID for S3 authentication
s3_endpoint	string	S3Endpoint specifies the S3-compatible storage endpoint

Property	Type	Description
<code>s3_force_path_style</code>	<code>boolean</code>	S3ForcePathStyle enables path-style S3 URLs (bucket.s3.amazonaws.com)
<code>s3_secret_access_key</code>	<code>string</code>	S3SecretAccessKey specifies the secret access key for S3 authentication
<code>s3_wal_path</code>	<code>string</code>	S3WalPath specifies the S3 path containing WAL files for PITR cloning
<code>timestamp</code>	<code>string</code>	EndTimestamp specifies the point-in-time to clone up to (RFC3339 format)
<code>uid</code>	<code>string</code>	UID specifies the unique identifier of the source cluster

.spec.clusterReplication

Description

ClusterReplication defines cross-cluster replication

Type

`object`

Property	Type	Description
<code>bootstrapSecret</code>	<code>string</code>	BootstrapSecret contains credentials for initial replication setup
<code>enabled</code>	<code>boolean</code>	Enabled controls whether cross-cluster replication is active
<code>isReplica</code>	<code>boolean</code>	IsReplica indicates if this cluster should act as a replica
<code>peerHost</code>	<code>string</code>	PeerHost specifies the hostname or IP of the peer cluster
<code>peerPort</code>	<code>integer</code>	PeerPort specifies the port number for connecting to the peer cluster
<code>replSvcType</code>	<code>string</code>	ReplSvcType defines the Kubernetes Service type for replication traffic
<code>syncMode</code>	<code>boolean</code>	SyncMode enables synchronous replication between clusters

`.spec.connectionPooler`

Description

ConnectionPooler configuration for connection pooling

Type

object

Property	Type	Description
<code>dockerImage</code>	<code>string</code>	DockerImage specifies the container image to use for the connection pooler. If not specified, uses the default image configured in the operator.
<code>maxDBConnections</code>	<code>integer</code>	MaxDBConnections defines the maximum number of database connections that the pooler will maintain to PostgreSQL. Defaults to 60 if not specified.
<code>mode</code>	<code>string</code>	Mode defines the connection pooling mode. Can be "session" or "transaction". In session mode, connections are held for the duration of a client session. In transaction mode, connections are returned to the pool after each transaction. Defaults to "transaction" if not specified.
<code>numberOfInstances</code>	<code>integer</code>	NumberOfInstances specifies how many connection pooler instances to run. If not set, defaults to 1 for PgBouncer or 2 for Pgpool-II.
<code>resources</code>	<code>object</code>	Resources defines CPU and memory requirements for the connection pooler containers. If not specified, uses operator defaults.

Property	Type	Description
<code>schema</code>	<code>string</code>	Schema specifies the database schema where connection pooler will operate. Defaults to "pooler" if not specified.
<code>user</code>	<code>string</code>	User specifies the database user that the connection pooler will use. Defaults to "pooler" if not specified.

`.spec.connectionPooler.resources`

Description

Resources defines CPU and memory requirements for the connection pooler containers. If not specified, uses operator defaults.

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	ResourceLimits defines the maximum resources allowed for containers
<code>requests</code>	<code>object</code>	ResourceRequests defines the minimum resources required for containers

`.spec.connectionPooler.resources.limits`

Description

ResourceLimits defines the maximum resources allowed for containers

Type

object

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

.spec.connectionPooler.resources.requests

Description

ResourceRequests defines the minimum resources required for containers

Type

object

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

.spec.databases

Description

Databases defines databases to be created in the cluster

Type

object

.spec.env

Description

Env defines environment variables for containers

Type

array

.spec.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 "".

Property	Type	Description
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.env[].valueFrom`

Description

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

Type

`object`

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

Property	Type	Description
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

`.spec.env[].valueFrom.fieldRef`

Description

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

Type

object

Required

fieldPath

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

.spec.env[].valueFrom.resourceFieldRef

Description

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

Type

object

Required

resource

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

`.spec.env[].valueFrom.secretKeyRef`

Description

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

Type

`object`

Required

`key`

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

Property	Type	Description
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

`.spec.exporter`

Description

Exporter defines Prometheus exporter configuration

Type

`object`

Property	Type	Description
<code>env</code>	<code>array</code>	

`.spec.exporter.env`

Type

`array`

`.spec.exporter.env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

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

`.spec.exporter.env[].valueFrom`

Description

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

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.

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

`.spec.exporter.env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.

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

`.spec.exporter.env[].valueFrom.fieldRef`

Description

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

Type

`object`

Required

`fieldPath`

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

`.spec.exporter.env[].valueFrom.resourceFieldRef`

Description

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

Type

object

Required

resource

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

.spec.exporter.env[].valueFrom.secretKeyRef

Description

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

Type

object

Required

key

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

`.spec.init_containers`

Description

InitContainersOld is a deprecated field for init containers

Type

`array`

`.spec.init_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 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
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 unique, and may not be defined in other sources.

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

Property	Type	Description
		Cannot be updated.
ports	array	List of ports to expose from the container. Not specifying here DOES NOT prevent that port from being exposed, which is listening on the default "0.0.0.0" address if the container will be accessible from the network. Modifying an array with strategic merge patch may corrupt the data. See https://github.com/kubernetes/kubernetes/issues/10926 for more information. Cannot be updated.
readinessProbe	object	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
resizePolicy	array	Resources resize policy for the container.
resources	object	Compute Resources required by this container. Cannot be updated. More info: https://kubernetes.io/docs/concepts/configuration/resource-limits/#resources-containers/
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers, this field is not specified, the restart behavior is defined by the Pod's restart policy.

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

Property	Type	Description
		container will always result in EOF. Default is false
stdinOnce	boolean	Whether the container runtime should close the stdin stream after it has been opened by a single attach. When the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process that starts before the first client attaches to stdin will never receive an EOF. Default is false
terminationMessagePath	string	Optional: Path at which the file to which the container termination message will be written is mounted into the container's filesystem. Message written is intended to be consumed by a human, so it should not be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited by the node. 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 lines, whichever is smaller. Defaults to File. Cannot be updated

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

`.spec.init_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.init_containers[].args[]`

Type

string

`.spec.init_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.init_containers[].command[]`

Type

string

`.spec.init_containers[].env`

Description

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

Type

array

`.spec.init_containers[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

name

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

.spec.init_containers[].env[].valueFrom

Description

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

Type

object

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

`.spec.init_containers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

`.spec.init_containers[].env[].valueFrom.fieldRef`

Description

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

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.init_containers[].env[].valueFrom.resourceFieldRef`

Description

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

Type

`object`

Required

`resource`

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

`.spec.init_containers[].env[].valueFrom.secretKeyRef`

Description

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

Type

object

Required

key

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

.spec.init_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.init_containers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

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

.spec.init_containers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

object

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

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

`.spec.init_containers[].envFrom[].secretRef`

Description

The Secret to select from

Type

`object`

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

`.spec.init_containers[].lifecycle`

Description

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

Type

`object`

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

`.spec.init_containers[].lifecycle.postStart`

Description

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

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

Type

`object`

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

`.spec.init_containers[].lifecycle.postStart.exec`

Description

Exec specifies the action to take.

Type

`object`

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

`.spec.init_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.init_containers[].lifecycle.postStart.exec.command[]`

Type

`string`

`.spec.init_containers[].lifecycle.postStart.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

port

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

.spec.init_containers[].lifecycle.postStart.httpGet.httpHea
ders

Description

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

Type

array

`.spec.init_containers[].lifecycle.postStart.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

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

`.spec.init_containers[].lifecycle.postStart.sleep`

Description

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

Type

object

Required

seconds

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

`.spec.init_containers[].lifecycle.postStart.tcpSocket`

Description

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

Type

`object`

Required

`port`

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

`.spec.init_containers[].lifecycle.preStop`

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace

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

Type

object

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

.spec.init_containers[].lifecycle.preStop.exec

Description

Exec specifies the action to take.

Type

object

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

`.spec.init_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.init_containers[].lifecycle.preStop.exec.command[]`

Type

`string`

`.spec.init_containers[].lifecycle.preStop.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

port

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

.spec.init_containers[].lifecycle.preStop.httpGet.httpHeaders

Description

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

Type

array

`.spec.init_containers[].lifecycle.preStop.httpGet.headers[]`

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.init_containers[].lifecycle.preStop.sleep`

Description

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

Type

object

Required

seconds

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

`.spec.init_containers[].lifecycle.preStop.tcpSocket`

Description

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

Type

`object`

Required

`port`

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

`.spec.init_containers[].livenessProbe`

Description

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

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRPC
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perform
<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/controllers/lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default value is 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. Default value is 1. Must be 1 for liveness and startup. Minimum value is 1.

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

`.spec.init_containers[].livenessProbe.exec`

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') 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.init_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.init_containers[].livenessProbe.exec.command[]`

Type

`string`

`.spec.init_containers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`

Required

port

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

.spec.init_containers[].livenessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

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

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

`.spec.init_containers[].livenessProbe.httpGet.httpHeaders`

Description

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

Type

`array`

`.spec.init_containers[].livenessProbe.httpGet.httpHeaders`

`[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name``value`

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

`.spec.init_containers[].livenessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

`object`

Required

`port`

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

`.spec.init_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.init_containers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

containerPort

Property	Type	Description
<code>containerPort</code>	<code>integer</code>	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.
<code>hostIP</code>	<code>string</code>	What host IP to bind the external port to.
<code>hostPort</code>	<code>integer</code>	Number of port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.

Property	Type	Description
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.init_containers[].readinessProbe`

Description

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

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

Type

`object`

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

Property	Type	Description
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perform
<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/lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default is 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. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCP Socket specifies an action involving a TCP port
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is killed. The grace period is the duration in seconds after the processes running on the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It specifies that the pod needs to terminate gracefully within this time by default. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no graceful termination is required).

Property	Type	Description
		(no opportunity to shut down). This is a beta requires enabling ProbeTerminationGracePeriodSeconds. Minimum value is 1. spec.terminationGracePeriodSeconds is use
timeoutSeconds	integer	Number of seconds after which the probe timeout. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes

.spec.init_containers[].readinessProbe.exec

Description

Exec specifies the action to take.

Type

object

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

.spec.init_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.init_containers[].readinessProbe.exec.command[]`****Type**`string`**`.spec.init_containers[].readinessProbe.grpc`****Description**

GRPC specifies an action involving a GRPC port.

Type`object`**Required**`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.

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

`.spec.init_containers[].readinessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

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

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

`.spec.init_containers[].readinessProbe.httpGet.httpHeaderS`

Description

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

Type

`array`

`.spec.init_containers[].readinessProbe.httpGet.httpHeaderS[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

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

`.spec.init_containers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

`object`

Required

`port`

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

`.spec.init_containers[].resizePolicy`

Description

Resources resize policy for the container.

Type

array

.spec.init_containers[].resizePolicy[]**Description**

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

resourceName

restartPolicy

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

.spec.init_containers[].resources**Description**

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

Type

object

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

.spec.init_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.init_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.init_containers[].securityContext

Description

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

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 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>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to

Property	Type	Description
		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 is DefaultProcMount which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to

Property	Type	Description
		start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that

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

`.spec.init_containers[].securityContext.capabilities`

Description

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

Type

`object`

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

`.spec.init_containers[].securityContext.capabilities.add`

Description

Added capabilities

Type

array

`.spec.init_containers[].securityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.init_containers[].securityContext.capabilities.drop`

Description

Removed capabilities

Type

array

`.spec.init_containers[].securityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.init_containers[].securityContext.seLinuxOptions`

Description

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

specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
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.init_containers[].securityContext.seccompProfile

Description

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

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	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.

`.spec.init_containers[].securityContext.windowsOptions`

Description

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

Type

`object`

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA

Property	Type	Description
		credential spec named by the GMSACredentialSpecName 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.init_containers[].startupProbe`

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted,

just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

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

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies an action involving a GRPC
<code>httpGet</code>	object	HTTPGet specifies the http request to perform
<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.

Property	Type	Description
<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>	TCPSocket specifies an action involving a TCP port. The port number is a required field in the <code>TCPSocket</code> object. The port number and the host name are optional fields in the <code>TCPSocket</code> object. If the host name is not specified, the default host name is <code>"localhost"</code> . The port number and the host name are optional fields in the <code>TCPSocket</code> object. If the host name is not specified, the default host name is <code>"localhost"</code> .
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is terminated. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It specifies that the container managed liveness probe of the pod to allow the grace period before being terminated. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. Defaults to 30 seconds. <code>spec.terminationGracePeriodSeconds</code> is used if not set.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. Maximum value is 30 seconds. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

`.spec.init_containers[].startupProbe.exec`

Description

Exec specifies the action to take.

Type

object

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

`.spec.init_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.init_containers[].startupProbe.exec.command[]`

Type

string

`.spec.init_containers[].startupProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

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

.spec.init_containers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

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

`.spec.init_containers[].startupProbe.httpGet.httpHeaders`

Description

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

Type

`array`

`.spec.init_containers[].startupProbe.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

namevalue

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.init_containers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

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

`.spec.init_containers[].volumeDevices`

Description

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

Type

array

`.spec.init_containers[].volumeDevices[]`

Description

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

Type

object

Required

devicePath

name

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

`.spec.init_containers[].volumeMounts`

Description

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

Type

array

.spec.init_containers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

mountPath

name

Property	Type	Description
<code>mountPath</code>	<code>string</code>	Path within the container at which the volume should be mounted. Must not contain ':'. <code>mountPropagation</code> 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.
<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>subPath</code>	<code>string</code>	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

Property	Type	Description
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

.spec.initContainers

Description

InitContainers defines initialization containers for the pod

Type

`array`

.spec.initContainers[]

Description

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

Type

`object`

Required

`name`

Property	Type	Description
<code>args</code>	<code>array</code>	Arguments to the entrypoint. The container image' if this is not provided. Variable references \$(VAR_I expanded using the container's environment. If a v be resolved, the reference in the input string will be

Property	Type	Description
		Double \$\$ are reduced to a single \$, which allows the \$(VAR_NAME) syntax: i.e. "\$\$(VAR_NAME)" v string literal "\$(VAR_NAME)". Escaped references expanded, regardless of whether the variable exists 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. References \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, 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. Can be updated.

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

Property	Type	Description
name	string	Name of the container specified as a DNS_LABEL container in a pod must have a unique name (DNS Cannot be updated.
ports	array	List of ports to expose from the container. Not spe here DOES NOT prevent that port from being exp which is listening on the default "0.0.0.0" address i container will be accessible from the network. Moc array with strategic merge patch may corrupt the d information See https://github.com/kubernetes/kubernetes/issues/1 Cannot be updated.
readinessProbe	object	Periodic probe of container service readiness. Cor removed from service endpoints if the probe fails. updated. More info: https://kubernetes.io/docs/concepts/workloads/poc lifecycle#container-probes
resizePolicy	array	Resources resize policy for the container.
resources	object	Compute Resources required by this container. Ca updated. More info: https://kubernetes.io/docs/concepts/configuration/r resources-containers/

Property	Type	Description
<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. The only allowed value is "Always". For non-init containers, if this field is not specified, the restart policy is defined by the Pod's restart policy and the container type. Setting RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted until all regular containers have terminated. Once all regular containers have completed, all init containers with "Always" will be shut down. This lifecycle differs from regular containers and is often referred to as a "sidecar" 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 completes or after any startupProbe has successfully completed.
<code>securityContext</code>	<code>object</code>	SecurityContext defines the security options the container will be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/
<code>startupProbe</code>	<code>object</code>	StartupProbe indicates that the Pod has successfully started. If specified, no other probes are executed until this probe successfully. If this probe fails, the Pod will be restarted. If the livenessProbe failed. This can be used to provide probe parameters at the beginning of a Pod's lifecycle. It might take a long time to load data or warm a cache before steady-state operation. This cannot be updated. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/

Property	Type	Description
		https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes ↗
<code>stdin</code>	<code>boolean</code>	Whether this container should allocate a buffer for container runtime. If this is not set, reads from stdin container will always result in EOF. Default is false
<code>stdinOnce</code>	<code>boolean</code>	Whether the container runtime should close the stdin after it has been opened by a single attach. When 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, remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process that starts from stdin will never receive an EOF. Default is false
<code>terminationMessagePath</code>	<code>string</code>	Optional: Path at which the file to which the container termination message will be written is mounted into the container's filesystem. Message written is intended to be consumed by Kubernetes. Can be <code>"/dev/termination-log"</code> , or <code>"/dev/stdout"</code> . Message will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited by the node's kubelet resources. Defaults to <code>/dev/termination-log</code> . Cannot be updated
<code>terminationMessagePolicy</code>	<code>string</code>	Indicate how the termination message should be populated. <code>File</code> 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 output if the termination message file is empty and

Property	Type	Description
		exited with an error. The log output is limited to 20 lines, whichever is smaller. Defaults to File. Cannot be updated.
<code>tty</code>	<code>boolean</code>	Whether this container should allocate a TTY for it. 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 for a container image. Cannot be updated.

`.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.initContainers[].args[]

Type

`string`

.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.initContainers[].command[]

Type

`string`

.spec.initContainers[].env

Description

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

Type

`array`

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

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

Description

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

Type

object

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

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

Description

Selects a key of a ConfigMap.

Type

object

Required

key

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

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

Description

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

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

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

Description

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

Type

`object`

Required

`resource`

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

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

Description

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

Type

object

Required

key

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

.spec.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.initContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

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

.spec.initContainers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

object

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

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

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

Description

The Secret to select from

Type

`object`

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

`.spec.initContainers[].lifecycle`

Description

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

Type

`object`

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

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

Description

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

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

Type

`object`

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

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

Description

Exec specifies the action to take.

Type

`object`

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

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

Type

`string`

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

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

port

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

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

Description

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

Type

array

`.spec.initContainers[].lifecycle.postStart.httpGet.headers[]`

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.initContainers[].lifecycle.postStart.sleep`

Description

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

Type

object

Required

seconds

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

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

Description

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

Type

`object`

Required

`port`

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

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

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace

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

Type

object

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

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

Description

Exec specifies the action to take.

Type

object

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

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

Type

`string`

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

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

port

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

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

Description

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

Type

array

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

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.initContainers[].lifecycle.preStop.sleep`

Description

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

Type

object

Required

seconds

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

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

Description

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

Type

`object`

Required

`port`

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

`.spec.initContainers[].livenessProbe`

Description

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

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies the action to take.
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies an action involving a GRPC
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perform
<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/lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default is 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. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.

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

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

Description

Exec specifies the action to take.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') 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.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.initContainers[].livenessProbe.exec.command[]`

Type

`string`

`.spec.initContainers[].livenessProbe.grpc`

Description

GRPC specifies an action involving a GRPC port.

Type

`object`

Required

port

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

.spec.initContainers[].livenessProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

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

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

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

Description

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

Type

`array`

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

name

value

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

.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

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

.spec.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.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.
<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.

Property	Type	Description
<code>name</code>	<code>string</code>	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
<code>protocol</code>	<code>string</code>	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

`.spec.initContainers[].readinessProbe`

Description

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

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

Type

`object`

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

Property	Type	Description
<code>httpGet</code>	<code>object</code>	HTTPGet specifies the http request to perform
<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/lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default is 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. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCP Socket specifies an action involving a TCP port
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is killed. The grace period is the duration in seconds after the processes running on the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It specifies that the pod needs to terminate gracefully within this time by default. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal.

Property	Type	Description
		(no opportunity to shut down). This is a beta requires enabling ProbeTerminationGracePeriodSeconds is use
timeoutSeconds	integer	Number of seconds after which the probe timeout Defaults to 1 second. Minimum value is 1. M https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes

.spec.initContainers[].readinessProbe.exec

Description

Exec specifies the action to take.

Type

object

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

.spec.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.initContainers[].readinessProbe.exec.command[]`****Type**`string`**`.spec.initContainers[].readinessProbe.grpc`****Description**

GRPC specifies an action involving a GRPC port.

Type`object`**Required**`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.

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

`.spec.initContainers[].readinessProbe.httpGet`

Description

HTTPGet specifies the http request to perform.

Type

`object`

Required

`port`

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

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

`.spec.initContainers[].readinessProbe.httpGet.httpHeaderS`

Description

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

Type

`array`

`.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
<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.initContainers[].readinessProbe.tcpSocket`

Description

TCPSocket specifies an action involving a TCP port.

Type

`object`

Required

`port`

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

`.spec.initContainers[].resizePolicy`

Description

Resources resize policy for the container.

Type

array

.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.initContainers[].resources

Description

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

Type

object

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.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.initContainers[].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.

.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.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.initContainers[].securityContext

Description

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

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 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>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to

Property	Type	Description
		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 is DefaultProcMount which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to

Property	Type	Description
		start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that

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

`.spec.initContainers[].securityContext.capabilities`

Description

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

Type

`object`

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

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

Description

Added capabilities

Type

array

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

Description

Capability represent POSIX capabilities type

Type

string

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

Description

Removed capabilities

Type

array

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

Description

Capability represent POSIX capabilities type

Type

string

`.spec.initContainers[].securityContext.seLinuxOptions`

Description

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

specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
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.initContainers[].securityContext.seccompProfile

Description

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

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	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.

`.spec.initContainers[].securityContext.windowsOptions`

Description

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

Type

`object`

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA

Property	Type	Description
		credential spec named by the GMSACredentialSpecName 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.initContainers[].startupProbe

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted,

just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info:

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

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies the action to take.
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies an action involving a GRPC
<code>httpGet</code>	object	HTTPGet specifies the http request to perform
<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.

Property	Type	Description
<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>	TCPSocket specifies an action involving a TCP port.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is killed. The grace period is the duration in seconds after the processes running on the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It specifies that the pod should terminate gracefully by this amount of time, by which time the process must have exited or been restarted. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. Defaults to 30 seconds. spec.terminationGracePeriodSeconds is used if specified.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. Maximum value is 30 seconds. https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

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

Description

Exec specifies the action to take.

Type

object

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

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

Type

string

.spec.initContainers[].startupProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

port

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

.spec.initContainers[].startupProbe.httpGet

Description

HTTPGet specifies the http request to perform.

Type

object

Required

port

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

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

Description

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

Type

`array`

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

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

namevalue

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.initContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies an action involving a TCP port.

Type

object

Required

port

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

`.spec.initContainers[].volumeDevices`

Description

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

Type

array

`.spec.initContainers[].volumeDevices[]`

Description

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

Type

object

Required

`devicePath`

`name`

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

`.spec.initContainers[].volumeMounts`

Description

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

Type

array

.spec.initContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

mountPath

name

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':'. mountPropagation 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.
mountPropagation	string	This must match the Name of a Volume.
name	string	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
readOnly	boolean	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPath	string	

Property	Type	Description
<code>subPathExpr</code>	<code>string</code>	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

.spec.maintenanceWindows

Description

MaintenanceWindows defines time windows for maintenance operations

Type

`array`

.spec.maintenanceWindows[]

Description

MaintenanceWindow describes the time window when the operator is allowed to do maintenance on a cluster.

Type

`object`

Property	Type	Description
<code>endTime</code>	<code>string</code>	EndTime defines the end of the maintenance window in HH:MM format

Property	Type	Description
<code>everyday</code>	<code>boolean</code>	Everyday enables maintenance every day of the week
<code>startTime</code>	<code>string</code>	StartTime defines the beginning of the maintenance window in HH:MM format

.spec.masterServiceAnnotations

Description

MasterServiceAnnotations defines annotations for master service

Type

`object`

.spec.nodeAffinity

Description

NodeAffinity defines node affinity rules for pod scheduling

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

Property	Type	Description
		the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to

Property	Type	Description
		eventually evict the pod from its node.

`.spec.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.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

preference

weight

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

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

Description

A node selector term, associated with the corresponding weight.

Type

`object`

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

`.spec.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

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

Type

array

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

Description

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

Type

object

Required

key operator

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

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

Type

string

.spec.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

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

Type

array

.spec.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

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

Type

object

Required

key operator

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

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

Type

string

.spec.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

object

Required

nodeSelectorTerms

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

.spec.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

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

Type

array

`.spec.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]`

Description

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

Type

object

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

`.spec.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

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

Type

array

`.spec.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

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

Type

object

Required

key operator

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

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

Type

string

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

Description

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

Type

array

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

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

Type

string

.spec.nodeSelector

Description

NodeSelector defines node labels for pod scheduling

Type

object

.spec.patroni

Description

Patroni configuration for cluster management

Type

object

Property	Type	Description
failsafe_mode	boolean	Enable failsafe mode for cluster
initdb	object	InitDB parameters for database initialization
loop_wait	integer	Loop wait time in seconds for leader election
maximum_lag_on_failover	number	Maximum lag in bytes allowed for failover (float32 because https://github.com/kubernetes/kubernetes/issues/30450)

Property	Type	Description
<code>pg_hba</code>	<code>array</code>	Custom <code>pg_hba.conf</code> entries
<code>retry_timeout</code>	<code>integer</code>	Retry timeout in seconds for operations
<code>slots</code>	<code>object</code>	Replication slots configuration
<code>synchronous_mode</code>	<code>boolean</code>	Enable synchronous replication
<code>synchronous_mode_strict</code>	<code>boolean</code>	Enable strict synchronous replication
<code>synchronous_node_count</code>	<code>integer</code>	Number of synchronous replicas
<code>ttl</code>	<code>integer</code>	Leader key TTL in seconds

`.spec.patroni.initdb`

Description

InitDB parameters for database initialization

Type

`object`

`.spec.patroni.pg_hba`

Description

Custom pg_hba.conf entries

Type

array

.spec.patroni.pg_hba[]

Type

string

.spec.patroni.slots

Description

Replication slots configuration

Type

object

.spec.pgpool2Settings

Description

Pgpool2Settings defines Pgpool-II configuration

Type

object

Property	Type	Description
<code>dockerImage</code>	<code>string</code>	DockerImage specifies the container image to use for Pgpool-II

Property	Type	Description
<code>maxPool</code>	<code>integer</code>	MaxPool defines the maximum number of cached connections per child process
<code>numInitChildren</code>	<code>integer</code>	NumInitChildren sets the number of preforked Pgpool-II server processes
<code>numberOfInstances</code>	<code>integer</code>	NumberOfInstances specifies how many Pgpool-II instances to run
<code>resources</code>	<code>object</code>	Resources specifies CPU and memory requirements for Pgpool-II containers
<code>user</code>	<code>string</code>	User specifies the database user for Pgpool-II connections

.spec.pgpool2Settings.resources

Description

Resources specifies CPU and memory requirements for Pgpool-II containers

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	ResourceLimits defines the maximum resources allowed for containers
<code>requests</code>	<code>object</code>	ResourceRequests defines the minimum resources required for containers

`.spec.pgpool2Settings.resources.limits`

Description

ResourceLimits defines the maximum resources allowed for containers

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

`.spec.pgpool2Settings.resources.requests`

Description

ResourceRequests defines the minimum resources required for containers

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

`.spec.podAnnotations`

Description

PodAnnotations defines annotations to add to pods

Type

`object`

`.spec.postgresql`

Description

PostgreSQL version and configuration parameters including: - PostgreSQL major version (e.g. "14") - Configuration parameters (postgresql.conf) - Configuration parameters (postgresql.conf) PostgreSQL version and configuration parameters

Type

`object`

Required

`version`

Property	Type	Description
<code>parameters</code>	<code>object</code>	PostgreSQL configuration parameters (postgresql.conf)

Property	Type	Description
<code>version</code>	<code>string</code>	PostgreSQL major version (e.g. "14")

`.spec.postgresql.parameters`

Description

PostgreSQL configuration parameters (postgresql.conf)

Type

`object`

`.spec.preparedDatabases`

Description

PreparedDatabases defines databases with pre-configured schemas and roles

Type

`object`

`.spec.repairOption`

Description

RepairOption defines options for cluster repair

Type

`object`

Property	Type	Description
<code>autoRecovery</code>	<code>boolean</code>	AutoRecovery enables automatic recovery attempts for failed clusters

Property	Type	Description
<code>startTimeout</code>	<code>integer</code>	StartTimeout defines the maximum time (in seconds) to wait for cluster startup before recovery

`.spec.replicaServiceAnnotations`

Description

ReplicaServiceAnnotations defines annotations for replica service

Type

`object`

`.spec.resources`

Description

Resource requests and limits for PostgreSQL containers

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	ResourceLimits defines the maximum resources allowed for containers
<code>requests</code>	<code>object</code>	ResourceRequests defines the minimum resources required for containers

`.spec.resources.limits`

Description

ResourceLimits defines the maximum resources allowed for containers

Type

object

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

.spec.resources.requests

Description

ResourceRequests defines the minimum resources required for containers

Type

object

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

.spec.serviceAnnotations

Description

ServiceAnnotations defines annotations to add to services

Type

object

.spec.serviceTemplates**Description**

ServiceTemplates defines custom service templates

Type

object

.spec.sidecars**Description**

Sidecars defines additional containers to run in the pod

Type

array

.spec.sidecars[]**Description**

Sidecar defines a container to be run in the same pod as the Postgres container.

Type

object

Property	Type	Description
env	array	Env defines environment variables to set in the sidecar container

Property	Type	Description
<code>image</code>	<code>string</code>	DockerImage specifies the container image to use for the sidecar
<code>name</code>	<code>string</code>	Name specifies the unique name of the sidecar container within the pod
<code>ports</code>	<code>array</code>	Ports defines the network ports to expose from the sidecar container
<code>resources</code>	<code>object</code>	Resources defines CPU and memory requirements for the sidecar container

`.spec.sidecars[].env`

Description

Env defines environment variables to set in the sidecar container

Type

`array`

`.spec.sidecars[].env[]`

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

name

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

.spec.sidecars[].env[].valueFrom

Description

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

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key of a ConfigMap.

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

`.spec.sidecars[].env[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.

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

`.spec.sidecars[].env[].valueFrom.fieldRef`

Description

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

Type

`object`

Required

`fieldPath`

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

`.spec.sidecars[].env[].valueFrom.resourceFieldRef`

Description

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

Type

object

Required

resource

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

`.spec.sidecars[].env[].valueFrom.secretKeyRef`

Description

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

Type

object

Required

key

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

`.spec.sidecars[].ports`

Description

Ports defines the network ports to expose from the sidecar container

Type

`array`

`.spec.sidecars[].ports[]`

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

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

`.spec.sidecars[].resources`

Description

Resources defines CPU and memory requirements for the sidecar container

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	ResourceLimits defines the maximum resources allowed for containers
<code>requests</code>	<code>object</code>	ResourceRequests defines the minimum resources required for containers

`.spec.sidecars[].resources.limits`

Description

ResourceLimits defines the maximum resources allowed for containers

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

`.spec.sidecars[].resources.requests`

Description

ResourceRequests defines the minimum resources required for containers

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	CPU requirement (e.g., "500m" or "1")
<code>memory</code>	<code>string</code>	Memory requirement (e.g., "512Mi" or "2Gi")

.spec.standby

Description

StandbyCluster defines configuration for standby clusters

Type

`object`

Property	Type	Description
<code>gs_wal_path</code>	<code>string</code>	GSWalPath specifies the Google Cloud Storage path for WAL archiving (e.g. "gs://bucket/path")
<code>s3_wal_path</code>	<code>string</code>	S3WalPath specifies the S3 path for WAL archiving (e.g. "s3://bucket/path")
<code>standby_host</code>	<code>string</code>	StandbyHost specifies the hostname or IP address of the primary cluster
<code>standby_port</code>	<code>string</code>	StandbyPort specifies the port number of the primary cluster

.spec.streams

Description

Streams defines configuration for streaming data

Type

array

.spec.streams[]

Description

Stream defines properties for creating FabricEventStream resources

Type

object

Required

applicationId

database

tables

Property	Type	Description
applicationId	string	ApplicationId is the unique identifier for the application producing the event stream
batchSize	integer	BatchSize controls how many events are batched together before being processed
database	string	Database specifies the PostgreSQL database where the event tables are located

Property	Type	Description
<code>filter</code>	<code>object</code>	Filter specifies optional filtering conditions for the event stream
<code>tables</code>	<code>object</code>	Tables defines the mapping of table names to their stream configurations

`.spec.streams[].filter`

Description

Filter specifies optional filtering conditions for the event stream

Type

`object`

`.spec.streams[].tables`

Description

Tables defines the mapping of table names to their stream configurations

Type

`object`

`.spec.tls`

Description

TLS defines TLS configuration for the cluster

Type

`object`

Property	Type	Description
<code>caFile</code>	<code>string</code>	CAFile specifies the path to the CA certificate file for client verification
<code>caSecretName</code>	<code>string</code>	CASecretName specifies the Kubernetes secret containing the CA certificate
<code>certificateFile</code>	<code>string</code>	CertificateFile specifies the path to the server certificate file
<code>privateKeyFile</code>	<code>string</code>	PrivateKeyFile specifies the path to the server private key file
<code>secretName</code>	<code>string</code>	SecretName specifies the Kubernetes secret containing TLS certificates

`.spec.tolerations`

Description

Tolerations defines pod tolerations for node taints

Type

`array`

`.spec.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.

Property	Type	Description
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.upgradeOption

Description

UpgradeOption defines upgrade options

Type

`object`

Property	Type	Description
<code>crVersion</code>	<code>string</code>	CRVersion specifies the target custom resource version for the upgrade

.spec.users

Description

Users defines additional PostgreSQL users and their roles

Type

`object`

.spec.usersCustomizedPasswd

Description

UsersCustomizedPasswd stores custom passwords for users

Type

`object`

.spec.usersWithInPlaceSecretRotation

Description

UsersWithInPlaceSecretRotation defines users requiring in-place secret rotation

Type

`array`

.spec.usersWithInPlaceSecretRotation[]

Type

`string`

.spec.usersWithSecretRotation

Description

UsersWithSecretRotation defines users requiring secret rotation

Type

`array`

.spec.usersWithSecretRotation[]

Type

`string`

.spec.volume

Description

Volume configuration for PostgreSQL data storage

Type

`object`

Required

size

Property	Type	Description
iops	integer	IOPS for provisioned IOPS SSD volumes
selector	object	Selector for matching existing PersistentVolumeClaims
size	string	Size of the volume (e.g., "100Gi")
storageClass	string	StorageClass name for dynamic provisioning
subPath	string	SubPath within the volume to mount
throughput	integer	Throughput for gp3 volumes (MiB/s)
type	string	VolumeType for EBS volumes (gp2, gp3, io1, etc.)

.spec.volume.selector

Description

Selector for matching existing PersistentVolumeClaims

Type

object

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

`.spec.volume.selector.matchExpressions`

Description

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

Type

`array`

`.spec.volume.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.volume.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.volume.selector.matchExpressions[].values[]

Type

string

.spec.volume.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

.status

Description

Status defines the observed state of the PostgreSQL cluster including current status, patroni state, and any error messages.

Type

object

Property	Type	Description
PostgresClusterStatus	string	PostgresClusterStatus indicates the current operational state of the PostgreSQL cluster
UsersCustomizedPasswdStatus	string	UsersCustomizedPasswdStatus tracks the status of custom password management for database users
clusterReplicationRole	string	ClusterReplicationRole indicates the replication role (primary/replica) of this cluster

Property	Type	Description
<code>message</code>	<code>string</code>	Message provides additional details about the cluster status or error conditions
<code>patroniStatus</code>	<code>object</code>	PatroniStatus contains status information for each Patroni-managed PostgreSQL instance
<code>upgradeStatus</code>	<code>object</code>	UpgradeStatus tracks version information for operator and custom resource upgrades

`.status.patroniStatus`

Description

PatroniStatus contains status information for each Patroni-managed PostgreSQL instance

Type

`object`

`.status.upgradeStatus`

Description

UpgradeStatus tracks version information for operator and custom resource upgrades

Type

`object`

Property	Type	Description
<code>crVersion</code>	<code>string</code>	CRVersion tracks the version of the custom resource specification
<code>operatorVersion</code>	<code>string</code>	OperatorVersion tracks the version of the operator that last processed this cluster

API Endpoints

The following API endpoints are available:

- `/apis/acid.zalan.do/v1/namespaces/{namespace}/postgresqls`
 - `DELETE` : delete collection of postgresql
 - `GET` : list objects of kind postgresql
 - `POST` : create a new postgresql
- `/apis/acid.zalan.do/v1/namespaces/{namespace}/postgresqls/{name}`
 - `DELETE` : delete the specified postgresql
 - `GET` : read the specified postgresql
 - `PATCH` : partially update the specified postgresql
 - `PUT` : replace the specified postgresql
- `/apis/acid.zalan.do/v1/namespaces/{namespace}/postgresqls/{name}/status`
 - `GET` : read status of the specified postgresql
 - `PATCH` : partially update status of the specified postgresql
 - `PUT` : replace status of the specified postgresql

/apis/acid.zalan.do/v1/namespaces/{namespace}/postgresqls

HTTP method

DELETE

Description

delete collection of postgresql

HTTP responses

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

HTTP method

GET

Description

list objects of kind postgresql

HTTP responses

HTTP code	Response body
200 - OK	postgresqlList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new postgresql

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>postgresql schema</code>	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>postgresql schema</code>
201 - Created	<code>postgresql schema</code>
202 - Accepted	<code>postgresql schema</code>

HTTP code	Response body
401 - Unauthorized	Empty

/apis/acid.zalan.do/v1/namespaces/{namespace}/postgresqls/{name}

HTTP method

[DELETE](#)

Description

delete the specified postgresql

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

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

HTTP method

[GET](#)

Description

read the specified postgresql

HTTP responses

HTTP code	Response body
200 - OK	<code>postgresql</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified postgresql

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>postgresql</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified postgresql

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
body	postgresql schema	application/json formatted

HTTP responses

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

/apis/acid.zalan.do/v1/namespaces/{namespace}/postgresqls/{name}/status

HTTP method

GET

Description

read status of the specified postgresql

HTTP responses

HTTP code	Response body
200 - OK	postgresql schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified postgresql

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>postgresql</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `postgresql`

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>postgresql schema</code>	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>postgresql schema</code>
201 - Created	<code>postgresql schema</code>
401 - Unauthorized	Empty



PostgresBackup

Description

PostgresBackup is the Schema for the postgresbackups API

Type

object

Specification

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

Property	Type	Description
<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>	PostgresBackupSpec defines the desired state of PostgresBackup
<code>status</code>	<code>object</code>	PostgresBackupStatus defines the observed state of PostgresBackup

.spec

Description

PostgresBackupSpec defines the desired state of PostgresBackup

Type

`object`

Required

`cluster`

Property	Type	Description
<code>cluster</code>	<code>string</code>	Cluster is the name of the PostgreSQL cluster to backup
<code>executeNode</code>	<code>string</code>	ExecuteNode specifies the node where the backup job should run If empty, the backup controller will automatically select a node

.status

Description

PostgresBackupStatus defines the observed state of PostgresBackup

Type

object

Property	Type	Description
backupName	string	BackupName is the unique name of this backup instance
clusterUid	string	ClusterUid is the unique identifier of the PostgreSQL cluster
configBackupStorage	object	ConfigBackupStorage contains the configuration for the backup storage
error	string	Error contains any error messages from the backup operation
executeNode	string	ExecuteNode is the node where the backup job is running
finishLsn	integer	FinishLsn is the log sequence number when the backup finished

Property	Type	Description
<code>finishTime</code>	<code>string</code>	FinishTime is when the backup operation completed
<code>lastModified</code>	<code>string</code>	LastModified is the timestamp when the backup was last modified
<code>pgVersion</code>	<code>string</code>	PGVersion is the PostgreSQL version of the cluster being backed up
<code>startLsn</code>	<code>integer</code>	StartLsn is the log sequence number when the backup started
<code>startTime</code>	<code>string</code>	StartTime is when the backup operation began
<code>state</code>	<code>string</code>	State represents the current state of the backup operation Possible values: "", "running", "failed", "succeeded", "deleteFailed"

.status.configBackupStorage

Description

ConfigBackupStorage contains the configuration for the backup storage

Type

`object`

Required

bucket

name

namespace

Property	Type	Description
bucket	string	
name	string	
namespace	string	

API Endpoints

The following API endpoints are available:

- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresbackups`
 - `DELETE` : delete collection of PostgresBackup
 - `GET` : list objects of kind PostgresBackup
 - `POST` : create a new PostgresBackup
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresbackups/{name}`
 - `DELETE` : delete the specified PostgresBackup
 - `GET` : read the specified PostgresBackup
 - `PATCH` : partially update the specified PostgresBackup
 - `PUT` : replace the specified PostgresBackup
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresbackups/{name}/status`
 - `GET` : read status of the specified PostgresBackup
 - `PATCH` : partially update status of the specified PostgresBackup
 - `PUT` : replace status of the specified PostgresBackup

/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresbackups

HTTP method

[DELETE](#)

Description

delete collection of PostgresBackup

HTTP responses

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

HTTP method

[GET](#)

Description

list objects of kind PostgresBackup

HTTP responses

HTTP code	Response body
200 - OK	PostgresBackupList schema
401 - Unauthorized	Empty

HTTP method

[POST](#)

Description

create a new PostgresBackup

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>PostgresBackup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresBackup</code> schema
201 - Created	<code>PostgresBackup</code> schema
202 - Accepted	<code>PostgresBackup</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresbackups/{name}

HTTP method

[DELETE](#)

Description

delete the specified PostgresBackup

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ✓ schema
202 - Accepted	Status ✓ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified PostgresBackup

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresBackup</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified PostgresBackup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresBackup</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified PostgresBackup

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>PostgresBackup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresBackup</code> schema
201 - Created	<code>PostgresBackup</code> schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresbackups/{name}/status

HTTP method

`GET`

Description

read status of the specified PostgresBackup

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresBackup</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified PostgresBackup

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>PostgresBackup</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `PostgresBackup`

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>PostgresBackup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresBackup</code> schema
201 - Created	<code>PostgresBackup</code> schema
401 - Unauthorized	Empty

PostgresRestore

Description

PostgresRestore is the Schema for the postgresrestores API

Type

object

Specification

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

Property	Type	Description
<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>	PostgresRestoreSpec defines the desired state of PostgresRestore
<code>status</code>	<code>object</code>	PostgresRestoreStatus defines the observed state of PostgresRestore

.spec

Description

PostgresRestoreSpec defines the desired state of PostgresRestore

Type

`object`

Required

`backupCluster`

`targetCluster`

`timestamp`

Property	Type	Description
<code>backupCluster</code>	<code>object</code>	BackupCluster contains details about the source backup cluster
<code>targetCluster</code>	<code>string</code>	TargetCluster is the name of the PostgreSQL cluster to restore into

Property	Type	Description
<code>timestamp</code>	<code>string</code>	Timestamp specifies the point-in-time to restore to in RFC3339 format

`.spec.backupCluster`

Description

BackupCluster contains details about the source backup cluster

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the source PostgreSQL cluster
<code>storage</code>	<code>object</code>	Storage configuration for accessing the backup
<code>uid</code>	<code>string</code>	Uid is the unique identifier of the source cluster

`.spec.backupCluster.storage`

Description

Storage configuration for accessing the backup

Type

`object`

Required

bucket

name

namespace

Property	Type	Description
bucket	string	Bucket name where backups are stored
name	string	Name of the storage configuration
namespace	string	Namespace where the storage configuration is located
s3option	object	S3Option contains S3-specific storage options

.spec.backupCluster.storage.s3Option

Description

S3Option contains S3-specific storage options

Type

object

Property	Type	Description
awsS3ForcePathStyle	string	AwsS3ForcePathStyle enables path-style S3 URLs (s3.amazonaws.com/BUCKET/KEY)
s3ForcePathStyle	boolean	S3ForcePathStyle forces path-style S3 URLs

Property	Type	Description
<code>useWalg</code>	<code>boolean</code>	UseWalg enables the WAL-G backup tool
<code>useWalgBackup</code>	<code>boolean</code>	UseWalgBackup enables WAL-G for backup operations
<code>walgDisableS3Sse</code>	<code>boolean</code>	WalgDisableS3Sse disables S3 server-side encryption for WAL-G

.status

Description

PostgresRestoreStatus defines the observed state of PostgresRestore

Type

`object`

Property	Type	Description
<code>error</code>	<code>string</code>	Error contains any error messages from the restore operation
<code>state</code>	<code>string</code>	State represents the current state of the restore operation

API Endpoints

The following API endpoints are available:

- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresrestores`
 - **DELETE** : delete collection of PostgresRestore
 - **GET** : list objects of kind PostgresRestore
 - **POST** : create a new PostgresRestore
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresrestores/{name}`
 - **DELETE** : delete the specified PostgresRestore
 - **GET** : read the specified PostgresRestore
 - **PATCH** : partially update the specified PostgresRestore
 - **PUT** : replace the specified PostgresRestore
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresrestores/{name}/status`
 - **GET** : read status of the specified PostgresRestore
 - **PATCH** : partially update status of the specified PostgresRestore
 - **PUT** : replace status of the specified PostgresRestore

`/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresrestores`

HTTP method

DELETE

Description

delete collection of PostgresRestore

HTTP responses

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

HTTP method

GET

Description

list objects of kind PostgresRestore

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresRestoreList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new PostgresRestore

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

Parameter	Type	Description
		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	PostgresRestore schema	application/json formatted

HTTP responses

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

/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresrestores/{name}

HTTP method

DELETE

Description

delete the specified PostgresRestore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified PostgresRestore

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresRestore</code>  schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified PostgresRestore

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>PostgresRestore</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `PostgresRestore`

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>PostgresRestore</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresRestore</code> schema
201 - Created	<code>PostgresRestore</code> schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/postgresrestores/{name}/status

HTTP method

GET

Description

read status of the specified PostgresRestore

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresRestore</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified PostgresRestore

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

Parameter	Type	Description
		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>PostgresRestore</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified PostgresRestore

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

Parameter	Type	Description
		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>PostgresRestore</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>PostgresRestore</code> schema
201 - Created	<code>PostgresRestore</code> schema
401 - Unauthorized	Empty