Cluster API Reference

Cluster API Core APIs

Cluster	ClusterClass	Machine
MachineSet	MachineDeployment	MachinePo
MachineHealthCheck	MachineDrainRule	

Cluster API Add-on APIs

ClusterResourceSet	ClusterResourceSetBinding

Cluster API IPAM APIs

IPAddress	IPAddressClaim

Cluster API Runtime APIs

ExtensionConfig

Kubeadm Bootstrap APIs

KubeadmConfig

KubeadmConfigTemplate

Kubeadm Control Plane APIs

KubeadmControlPlane

KubeadmControlPlaneTemplate

Cluster API Core APIs

Cluster	ClusterClass	Machine
MachineSet	MachineDeployment	MachinePo
MachineHealthCheck	MachineDrainRule	

Cluster [cluster.x-k8s.io/v1beta1]

Description

Cluster is the Schema for the clusters API.

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of Cluster.
`status`	`object`	status is the observed state of Cluster.

.spec

Description

spec is the desired state of Cluster.

Type

`object`

Property	Type	Description
`availabilityGates`	`array`	availabilityGates specifies additional conditions to include when evaluating Cluster Available condition.

Property	Type	Description
		If this field is not defined and the Cluster implements a managed topology, availabilityGates from the corresponding ClusterClass will be used, if any. NOTE: this field is considered only for computing v1beta2 conditions.
`clusterNetwork`	`object`	clusterNetwork represents the cluster network configuration.
`controlPlaneEndpoint`	`object`	controlPlaneEndpoint represents the endpoint used to communicate with the control plane.
`controlPlaneRef`	`object`	controlPlaneRef is an optional reference to a provider-specific resource that holds the details for provisioning the Control Plane for a Cluster.
`infrastructureRef`	`object`	infrastructureRef is a reference to a provider-specific resource that holds the details for provisioning infrastructure for a cluster in said provider.
`paused`	`boolean`	paused can be used to prevent controllers from processing the Cluster and all its associated objects.

Property	Type	Description
`topology`	`object`	topology encapsulates the topology for the cluster. NOTE: It is required to enable the ClusterTopology feature gate flag to activate managed topologies support; this feature is highly experimental, and parts of it might still be not implemented.

.spec.availabilityGates

Description

availabilityGates specifies additional conditions to include when evaluating Cluster Available condition. If this field is not defined and the Cluster implements a managed topology, availabilityGates from the corresponding ClusterClass will be used, if any. NOTE: this field is considered only for computing v1beta2 conditions.

Type

`array`

.spec.availabilityGates[]

Description

ClusterAvailabilityGate contains the type of a Cluster condition to be used as availability gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Cluster's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as availability gates.
`polarity`	`string`	polarity of the conditionType specified in this availabilityGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.clusterNetwork

Description

clusterNetwork represents the cluster network configuration.

Type

`object`

Property	Type	Description
`apiServerPort`	`integer`	apiServerPort specifies the port the API Server should bind to. Defaults to 6443.

Property	Type	Description
`pods`	`object`	pods is the network ranges from which Pod networks are allocated.
`serviceDomain`	`string`	serviceDomain is the domain name for services.
`services`	`object`	services is the network ranges from which service VIPs are allocated.

.spec.clusterNetwork.pods

Description

pods is the network ranges from which Pod networks are allocated.

Type

`object`

Required

`cidrBlocks`

Property	Type	Description
`cidrBlocks`	`array`	cidrBlocks is a list of CIDR blocks.

.spec.clusterNetwork.pods.cidrBlocks

Description

cidrBlocks is a list of CIDR blocks.

Type

`array`

.spec.clusterNetwork.pods.cidrBlocks[]

Type

`string`

.spec.clusterNetwork.services

Description

services is the network ranges from which service VIPs are allocated.

Type

`object`

Required

`cidrBlocks`

Property	Type	Description
`cidrBlocks`	`array`	cidrBlocks is a list of CIDR blocks.

.spec.clusterNetwork.services.cidrBlocks

Description

cidrBlocks is a list of CIDR blocks.

Type

`array`

.spec.clusterNetwork.services.cidrBlocks[]

Type

`string`

.spec.controlPlaneEndpoint

Description

controlPlaneEndpoint represents the endpoint used to communicate with the control plane.

Type

`object`

Required

`host` `port`

Property	Type	Description
`host`	`string`	host is the hostname on which the API server is serving.
`port`	`integer`	port is the port on which the API server is serving.

.spec.controlPlaneRef

Description

controlPlaneRef is an optional reference to a provider-specific resource that holds the details for provisioning the Control Plane for a Cluster.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.

Property	Type	Description
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-

Property	Type	Description
		architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.infrastructureRef

Description

infrastructureRef is a reference to a provider-specific resource that holds the details for provisioning infrastructure for a cluster in said provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2

Property	Type	Description
		in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.topology

Description

topology encapsulates the topology for the cluster. NOTE: It is required to enable the ClusterTopology feature gate flag to activate managed topologies support; this feature is highly experimental, and parts of it might still be not implemented.

Type

`object`

Required

`class` `version`

Property	Type	Description
`class`	`string`	class is the name of the ClusterClass object to create the topology.
`classNamespace`	`string`	classNamespace is the namespace of the ClusterClass that should be used for the topology. If classNamespace is empty or not set, it is defaulted to the namespace of the Cluster object. classNamespace must be a valid namespace name and because of that be at most 63 characters in length and it must consist only of lower case alphanumeric characters or hyphens (-), and must start and end with an alphanumeric character.
`controlPlane`	`object`	controlPlane describes the cluster control plane.
`rolloutAfter`	`string`	rolloutAfter performs a rollout of the entire cluster one component at a time, control plane first and then machine deployments.

Property	Type	Description
		Deprecated: This field has no function and is going to be removed in the next apiVersion.
`variables`	`array`	variables can be used to customize the Cluster through patches. They must comply to the corresponding VariableClasses defined in the ClusterClass.
`version`	`string`	version is the Kubernetes version of the cluster.
`workers`	`object`	workers encapsulates the different constructs that form the worker nodes for the cluster.

.spec.topology.controlPlane

Description

controlPlane describes the cluster control plane.

Type

`object`

Property	Type	Description
`machineHealthCheck`	`object`	machineHealthCheck allows to enable, disable and override the MachineHealthCheck configuration in the ClusterClass for this control plane.

Property	Type	Description
`metadata`	`ObjectMeta`	metadata is the metadata applied to the ControlPlane and the Machines of the ControlPlane if the ControlPlaneTemplate referenced by the ClusterClass is machine based. If not, it is applied only to the ControlPlane. At runtime this metadata is merged with the corresponding metadata from the ClusterClass.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.

Property	Type	Description
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. If this field is not defined, readinessGates from the corresponding ControlPlaneClass will be used, if any. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: Specific control plane provider implementations might automatically extend the list of readinessGates; e.g. the kubeadm control provider adds ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc.

Property	Type	Description
`replicas`	`integer`	replicas is the number of control plane nodes. If the value is nil, the ControlPlane object is created without the number of Replicas and it's assumed that the control plane controller does not implement support for this field. When specified against a control plane provider that lacks support for this field, this value will be ignored.
`variables`	`object`	variables can be used to customize the ControlPlane through patches.

.spec.topology.controlPlane.machineHealthCheck

Description

machineHealthCheck allows to enable, disable and override the MachineHealthCheck configuration in the ClusterClass for this control plane.

Type

`object`

Property	Type	Description
`enable`	`boolean`	enable controls if a MachineHealthCheck should be created for the target machines. If false: No MachineHealthCheck will be created. If not set(default): A MachineHealthCheck will be created if it is defined here or in the

Property	Type	Description
		associated ClusterClass. If no MachineHealthCheck is defined then none will be created. If true: A MachineHealthCheck is guaranteed to be created. Cluster validation will block if `enable` is true and no MachineHealthCheck definition is available.
`maxUnhealthy`	`integer|string`	maxUnhealthy specifies the maximum number of unhealthy machines allowed. Any further remediation is only allowed if at most "maxUnhealthy" machines selected by "selector" are not healthy.
`nodeStartupTimeout`	`string`	nodeStartupTimeout allows to set the maximum time for MachineHealthCheck to consider a Machine unhealthy if a corresponding Node isn't associated through a `Spec.ProviderID` field. The duration set in this field is compared to the greatest of: • Cluster's infrastructure ready condition timestamp (if and when available) • Control Plane's initialized condition timestamp (if and when available) • Machine's infrastructure ready condition timestamp (if and when available) • Machine's metadata creation timestamp

Property	Type	Description
		Defaults to 10 minutes. If you wish to disable this feature, set the value explicitly to 0.
`remediationTemplate`	`object`	remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.
`unhealthyConditions`	`array`	unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.
`unhealthyRange`	`string`	unhealthyRange specifies the range of unhealthy machines allowed. Any further remediation is only allowed if the number of machines selected by "selector" as not healthy is within the range of "unhealthyRange". Takes precedence over maxUnhealthy. Eg. "[3-5]" - This means that remediation will be allowed only when: (a) there are at least 3 unhealthy machines (and) (b) there are at most 5 unhealthy machines

.spec.topology.controlPlane.machineHealthCheck.remediationTemplate

Description

remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.topology.controlPlane.machineHealthCheck.unhealthyConditions

Description

unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.

Type

`array`

.spec.topology.controlPlane.machineHealthCheck.unhealthyConditions[]

Description

UnhealthyCondition represents a Node condition type and value with a timeout specified as a duration. When the named condition has been in the given status for at least the timeout value, a node is considered unhealthy.

Type

`object`

Required

`status` `timeout` `type`

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`timeout`	`string`	timeout is the duration that a node must be in a given status for, after which the node is considered unhealthy. For example, with a value of "1h", the node must match the status for at least 1 hour before being considered unhealthy.
`type`	`string`	type of Node condition

.spec.topology.controlPlane.readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external

controllers, reporting the status of special software/hardware installed on the Machine. If this field is not defined, readinessGates from the corresponding ControlPlaneClass will be used, if any. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: Specific control plane provider implementations might automatically extend the list of readinessGates; e.g. the kubeadm control provider adds ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc.

Type

`array`

.spec.topology.controlPlane.readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A

Property	Type	Description
		negative polarity means that the condition should report a false status under normal conditions.

.spec.topology.controlPlane.variables

Description

variables can be used to customize the ControlPlane through patches.

Type

`object`

Property	Type	Description
`overrides`	`array`	overrides can be used to override Cluster level variables.

.spec.topology.controlPlane.variables.overrides

Description

overrides can be used to override Cluster level variables.

Type

`array`

.spec.topology.controlPlane.variables.overrides[]

Description

ClusterVariable can be used to customize the Cluster through patches. Each ClusterVariable is associated with a Variable definition in the ClusterClass `status` variables.

Type

`object`

Required

`name` `value`

Property	Type	Description
`definitionFrom`	`string`	definitionFrom specifies where the definition of this Variable i Deprecated: This field is deprecated, must not be set anymo in the next apiVersion.
`name`	`string`	name of the variable.
`value`		value of the variable. Note: the value will be validated agains corresponding ClusterClassVariable from the ClusterClass. N apiextensionsv1.JSON instead of a custom JSON type, beca coded schema for apiextensionsv1.JSON which cannot be p controller-tools, i.e. it is not possible to have no type field. Re https://github.com/kubernetes-sigs/controller-tools/blob/d0e03a142d0ecdd5491593e941ee1d6b5d91dba6L111 ↗

.spec.topology.variables

Description

variables can be used to customize the Cluster through patches. They must comply to the corresponding VariableClasses defined in the ClusterClass.

Type

`array`

.spec.topology.variables[]

Description

ClusterVariable can be used to customize the Cluster through patches. Each ClusterVariable is associated with a Variable definition in the ClusterClass `status` variables.

Type

`object`

Required

`name` `value`

Property	Type	Description
`definitionFrom`	`string`	definitionFrom specifies where the definition of this Variable i Deprecated: This field is deprecated, must not be set anymo in the next apiVersion.
`name`	`string`	name of the variable.
`value`		value of the variable. Note: the value will be validated agains corresponding ClusterClassVariable from the ClusterClass. N apiextensionsv1.JSON instead of a custom JSON type, beca coded schema for apiextensionsv1.JSON which cannot be p controller-tools, i.e. it is not possible to have no type field. Re https://github.com/kubernetes-sigs/controller-tools/blob/d0e03a142d0ecdd5491593e941ee1d6b5d91dba6 L111 ↗

.spec.topology.workers

Description

workers encapsulates the different constructs that form the worker nodes for the cluster.

Type

`object`

Property	Type	Description
`machineDeployments`	`array`	machineDeployments is a list of machine deployments in the cluster.
`machinePools`	`array`	machinePools is a list of machine pools in the cluster.

.spec.topology.workers.machineDeployments

Description

machineDeployments is a list of machine deployments in the cluster.

Type

`array`

.spec.topology.workers.machineDeployments[]

Description

MachineDeploymentTopology specifies the different parameters for a set of worker nodes in the topology. This set of nodes is managed by a MachineDeployment object whose lifecycle is managed by the Cluster controller.

Type

`object`

Required

`class` `name`

Property	Type	Description
`class`	`string`	class is the name of the MachineDeploymentClass used to create the set of worker nodes. This should match one of the deployment classes defined in the ClusterClass object mentioned in the `Cluster.Spec.Class` field.
`failureDomain`	`string`	failureDomain is the failure domain the machines will be created in. Must match a key in the FailureDomains map stored on the cluster object.
`machineHealthCheck`	`object`	machineHealthCheck allows to enable, disable and override the MachineHealthCheck configuration in the ClusterClass for this MachineDeployment.
`metadata`	`ObjectMeta`	metadata is the metadata applied to the MachineDeployment and the machines of the MachineDeployment. At runtime this metadata is merged with the corresponding metadata from the ClusterClass.

Property	Type	Description
`minReadySeconds`	`integer`	minReadySeconds is the minimum number of seconds for which a newly created machine should be ready. Defaults to 0 (machine will be considered available as soon as it is ready)
`name`	`string`	name is the unique identifier for this MachineDeploymentTopology. The value is used with other unique identifiers to create a MachineDeployment's Name (e.g. cluster's name, etc). In case the name is greater than the allowed maximum length, the values are hashed together.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend

Property	Type	Description
		on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. If this field is not defined, readinessGates from the corresponding MachineDeploymentClass will be used, if any. NOTE: This field is considered only for computing v1beta2 conditions.
`replicas`	`integer`	replicas is the number of worker nodes belonging to this set. If the value is nil, the MachineDeployment is created without the number of Replicas (defaulting to 1) and it's assumed that an external entity (like cluster autoscaler) is responsible for the management of this value.

Property	Type	Description
`strategy`	`object`	strategy is the deployment strategy to use to replace existing machines with new ones.
`variables`	`object`	variables can be used to customize the MachineDeployment through patches.

.spec.topology.workers.machineDeployments[].machineHealthCheck

Description

machineHealthCheck allows to enable, disable and override the MachineHealthCheck configuration in the ClusterClass for this MachineDeployment.

Type

`object`

Property	Type	Description
`enable`	`boolean`	enable controls if a MachineHealthCheck should be created for the target machines. If false: No MachineHealthCheck will be created. If not set(default): A MachineHealthCheck will be created if it is defined here or in the associated ClusterClass. If no MachineHealthCheck is defined then none will be created.

Property	Type	Description
		If true: A MachineHealthCheck is guaranteed to be created. Cluster validation will block if `enable` is true and no MachineHealthCheck definition is available.
`maxUnhealthy`	`integer|string`	maxUnhealthy specifies the maximum number of unhealthy machines allowed. Any further remediation is only allowed if at most "maxUnhealthy" machines selected by "selector" are not healthy.
`nodeStartupTimeout`	`string`	nodeStartupTimeout allows to set the maximum time for MachineHealthCheck to consider a Machine unhealthy if a corresponding Node isn't associated through a `Spec.ProviderID` field. The duration set in this field is compared to the greatest of: • Cluster's infrastructure ready condition timestamp (if and when available) • Control Plane's initialized condition timestamp (if and when available) • Machine's infrastructure ready condition timestamp (if and when available) • Machine's metadata creation timestamp Defaults to 10 minutes. If you wish to disable this feature, set the value explicitly to 0.

Property	Type	Description
`remediationTemplate`	`object`	remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.
`unhealthyConditions`	`array`	unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.
`unhealthyRange`	`string`	unhealthyRange specifies the range of unhealthy machines allowed. Any further remediation is only allowed if the number of machines selected by "selector" as not healthy is within the range of "unhealthyRange". Takes precedence over maxUnhealthy. Eg. "[3-5]" - This means that remediation will be allowed only when: (a) there are at least 3 unhealthy machines (and) (b) there are at most 5 unhealthy machines

.spec.topology.workers.machineDeployments[].machineHealthCheck.remediationTemplate

Description

remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.topology.workers.machineDeployments[].machineHealthCheck.unhealthyConditions

Description

unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.

Type

`array`

.spec.topology.workers.machineDeployments[].machineHealthCheck.unhealthyConditions[]

Description

UnhealthyCondition represents a Node condition type and value with a timeout specified as a duration. When the named condition has been in the given status for at least the timeout value, a node is considered unhealthy.

Type

`object`

Required

`status` `timeout` `type`

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`timeout`	`string`	timeout is the duration that a node must be in a given status for, after which the node is considered unhealthy. For example, with a value of "1h", the node must match the status for at least 1 hour before being considered unhealthy.
`type`	`string`	type of Node condition

.spec.topology.workers.machineDeployments[].readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. If this field is not defined, readinessGates from the corresponding MachineDeploymentClass will be used, if any. NOTE: This field is considered only for computing v1beta2 conditions.

Type

`array`

.spec.topology.workers.machineDeployments[].readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition

Property	Type	Description
		should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.topology.workers.machineDeployments[].strategy

Description

strategy is the deployment strategy to use to replace existing machines with new ones.

Type

`object`

Property	Type	Description
`remediation`	`object`	remediation controls the strategy of remediating unhealthy machines and how remediating operations should occur during the lifecycle of the dependant MachineSets.
`rollingUpdate`	`object`	rollingUpdate is the rolling update config params. Present only if MachineDeploymentStrategyType = RollingUpdate.
`type`	`string`	type of deployment. Allowed values are RollingUpdate and OnDelete. The default is RollingUpdate.

.spec.topology.workers.machineDeployments[].strategy.remediation

Description

remediation controls the strategy of remediating unhealthy machines and how remediating operations should occur during the lifecycle of the dependant MachineSets.

Type

`object`

Property	Type	Description
`maxInFlight`	`integer|string`	maxInFlight determines how many in flight remediations should happen at the same time. Remediation only happens on the MachineSet with the most current revision, while older MachineSets (usually present during rollout operations) aren't allowed to remediate. Note: In general (independent of remediations), unhealthy machines are always prioritized during scale down operations over healthy ones. MaxInFlight can be set to a fixed number or a percentage. Example: when this is set to 20%, the MachineSet controller deletes at most 20% of the desired replicas. If not set, remediation is limited to all machines (bounded by replicas) under the active MachineSet's management.

.spec.topology.workers.machineDeployments[].strategy.rollingUpdate

Description

rollingUpdate is the rolling update config params. Present only if MachineDeploymentStrategyType = RollingUpdate.

Type

`object`

Property	Type	Description
`deletePolicy`	`string`	deletePolicy defines the policy used by the MachineDeployment to identify nodes to delete when downscaling. Valid values are "Random", "Newest", "Oldest" When no value is supplied, the default DeletePolicy of MachineSet is used
`maxSurge`	`integer|string`	maxSurge is the maximum number of machines that can be scheduled above the desired number of machines. Value can be an absolute number (ex: 5) or a percentage of desired machines (ex: 10%). This can not be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up. Defaults to 1. Example: when this is set to 30%, the new MachineSet can be scaled up immediately when the rolling update starts, such that the total number of old and new machines do not exceed 130% of desired machines. Once old machines have been killed, new MachineSet can be scaled up further, ensuring that total number of machines running at any time during the update is at most 130% of desired machines.
`maxUnavailable`	`integer|string`	maxUnavailable is the maximum number of machines that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of desired machines (ex: 10%). Absolute

Property	Type	Description
		number is calculated from percentage by rounding down. This can not be 0 if MaxSurge is 0. Defaults to 0. Example: when this is set to 30%, the old MachineSet can be scaled down to 70% of desired machines immediately when the rolling update starts. Once new machines are ready, old MachineSet can be scaled down further, followed by scaling up the new MachineSet, ensuring that the total number of machines available at all times during the update is at least 70% of desired machines.

.spec.topology.workers.machineDeployments[].variables

Description

variables can be used to customize the MachineDeployment through patches.

Type

`object`

Property	Type	Description
`overrides`	`array`	overrides can be used to override Cluster level variables.

.spec.topology.workers.machineDeployments[].variables.overrides

Description

overrides can be used to override Cluster level variables.

Type

`array`

.spec.topology.workers.machineDeployments[].variables.overrides[]

Description

ClusterVariable can be used to customize the Cluster through patches. Each ClusterVariable is associated with a Variable definition in the ClusterClass `status` variables.

Type

`object`

Required

`name` `value`

Property	Type	Description
`definitionFrom`	`string`	definitionFrom specifies where the definition of this Variable i Deprecated: This field is deprecated, must not be set anymo in the next apiVersion.
`name`	`string`	name of the variable.

Property	Type	Description
`value`		value of the variable. Note: the value will be validated agains corresponding ClusterClassVariable from the ClusterClass. N apiextensionsv1.JSON instead of a custom JSON type, beca coded schema for apiextensionsv1.JSON which cannot be p controller-tools, i.e. it is not possible to have no type field. Re https://github.com/kubernetes-sigs/controller-tools/blob/d0e03a142d0ecdd5491593e941ee1d6b5d91dba6 L111 ↗

.spec.topology.workers.machinePools

Description

machinePools is a list of machine pools in the cluster.

Type

`array`

.spec.topology.workers.machinePools[]

Description

MachinePoolTopology specifies the different parameters for a pool of worker nodes in the topology. This pool of nodes is managed by a MachinePool object whose lifecycle is managed by the Cluster controller.

Type

`object`

Required

`class` `name`

Property	Type	Description
`class`	`string`	class is the name of the MachinePoolClass used to create the pool of worker nodes. This should match one of the deployment classes defined in the ClusterClass object mentioned in the `Cluster.Spec.Class` field.
`failureDomains`	`array`	failureDomains is the list of failure domains the machine pool will be created in. Must match a key in the FailureDomains map stored on the cluster object.
`metadata`	`ObjectMeta`	metadata is the metadata applied to the MachinePool. At runtime this metadata is merged with the corresponding metadata from the ClusterClass.
`minReadySeconds`	`integer`	minReadySeconds is the minimum number of seconds for which a newly created machine pool should be ready. Defaults to 0 (machine will be considered available as soon as it is ready)

Property	Type	Description
`name`	`string`	name is the unique identifier for this MachinePoolTopology. The value is used with other unique identifiers to create a MachinePool's Name (e.g. cluster's name, etc). In case the name is greater than the allowed maximum length, the values are hashed together.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the MachinePool hosts after the MachinePool is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.

Property	Type	Description
`replicas`	`integer`	replicas is the number of nodes belonging to this pool. If the value is nil, the MachinePool is created without the number of Replicas (defaulting to 1) and it's assumed that an external entity (like cluster autoscaler) is responsible for the management of this value.
`variables`	`object`	variables can be used to customize the MachinePool through patches.

.spec.topology.workers.machinePools[].failureDomains

Description

failureDomains is the list of failure domains the machine pool will be created in. Must match a key in the FailureDomains map stored on the cluster object.

Type

`array`

.spec.topology.workers.machinePools[].failureDomains[]

Type

`string`

.spec.topology.workers.machinePools[].variables

Description

variables can be used to customize the MachinePool through patches.

Type

`object`

Property	Type	Description
`overrides`	`array`	overrides can be used to override Cluster level variables.

.spec.topology.workers.machinePools[].variables.overrides

Description

overrides can be used to override Cluster level variables.

Type

`array`

.spec.topology.workers.machinePools[].variables.overrides[]

Description

ClusterVariable can be used to customize the Cluster through patches. Each ClusterVariable is associated with a Variable definition in the ClusterClass `status` variables.

Type

`object`

Required

`name` `value`

Property	Type	Description
`definitionFrom`	`string`	definitionFrom specifies where the definition of this Variable i

Property	Type	Description
		Deprecated: This field is deprecated, must not be set anymo in the next apiVersion.
`name`	`string`	name of the variable.
`value`		value of the variable. Note: the value will be validated agains corresponding ClusterClassVariable from the ClusterClass. N apiextensionsv1.JSON instead of a custom JSON type, beca coded schema for apiextensionsv1.JSON which cannot be p controller-tools, i.e. it is not possible to have no type field. Re https://github.com/kubernetes-sigs/controller-tools/blob/d0e03a142d0ecdd5491593e941ee1d6b5d91dba6 L111 ↗

.status

Description

status is the observed state of Cluster.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions defines current service state of the cluster.
`controlPlaneReady`	`boolean`	controlPlaneReady denotes if the control plane became ready during initial provisioning to

Property	Type	Description
		receive requests. NOTE: this field is part of the Cluster API contract and it is used to orchestrate provisioning. The value of this field is never updated after provisioning is completed. Please use conditions to check the operational state of the control plane.
`failureDomains`	`object`	failureDomains is a slice of failure domain objects synced from the infrastructure provider.
`failureMessage`	`string`	failureMessage indicates that there is a fatal problem reconciling the state, and will be set to a descriptive error message. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`failureReason`	`string`	failureReason indicates that there is a fatal problem reconciling the state, and will be set to a token value suitable for programmatic interpretation. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.

Property	Type	Description
`infrastructureReady`	`boolean`	infrastructureReady is the state of the infrastructure provider.
`observedGeneration`	`integer`	observedGeneration is the latest generation observed by the controller.
`phase`	`string`	phase represents the current phase of cluster actuation.
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in Cluster's status with the V1Beta2 version.

.status.conditions

Description

conditions defines current service state of the cluster.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.failureDomains

Description

failureDomains is a slice of failure domain objects synced from the infrastructure provider.

Type

`object`

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in Cluster's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions represents the observations of a Cluster's current state. Known condition types are Available, InfrastructureReady, ControlPlaneInitialized, ControlPlaneAvailable, WorkersAvailable, MachinesReady MachinesUpToDate,

Property	Type	Description
		RemoteConnectionProbe, ScalingUp, ScalingDown, Remediating, Deleting, Paused. Additionally, a TopologyReconciled condition will be added in case the Cluster is referencing a ClusterClass / defining a managed Topology.
`controlPlane`	`object`	controlPlane groups all the observations about Cluster's ControlPlane current state.
`workers`	`object`	workers groups all the observations about Cluster's Workers current state.

.status.v1beta2.conditions

Description

conditions represents the observations of a Cluster's current state. Known condition types are Available, InfrastructureReady, ControlPlaneInitialized, ControlPlaneAvailable, WorkersAvailable, MachinesReady MachinesUpToDate, RemoteConnectionProbe, ScalingUp, ScalingDown, Remediating, Deleting, Paused. Additionally, a TopologyReconciled condition will be added in case the Cluster is referencing a ClusterClass / defining a managed Topology.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.

Property	Type	Description
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

.status.v1beta2.controlPlane

Description

controlPlane groups all the observations about Cluster's ControlPlane current state.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the total number of available control plane machines in this cluster. A machine is considered available when Machine's Available condition is true.

Property	Type	Description
`desiredReplicas`	`integer`	desiredReplicas is the total number of desired control plane machines in this cluster.
`readyReplicas`	`integer`	readyReplicas is the total number of ready control plane machines in this cluster. A machine is considered ready when Machine's Ready condition is true.
`replicas`	`integer`	replicas is the total number of control plane machines in this cluster. NOTE: replicas also includes machines still being provisioned or being deleted.
`upToDateReplicas`	`integer`	upToDateReplicas is the number of up-to-date control plane machines in this cluster. A machine is considered up-to-date when Machine's UpToDate condition is true.

.status.v1beta2.workers

Description

workers groups all the observations about Cluster's Workers current state.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the total number of available worker machines in this cluster. A machine is considered available when Machine's Available condition is true.
`desiredReplicas`	`integer`	desiredReplicas is the total number of desired worker machines in this cluster.
`readyReplicas`	`integer`	readyReplicas is the total number of ready worker machines in this cluster. A machine is considered ready when Machine's Ready condition is true.
`replicas`	`integer`	replicas is the total number of worker machines in this cluster. NOTE: replicas also includes machines still being provisioned or being deleted.
`upToDateReplicas`	`integer`	upToDateReplicas is the number of up-to-date worker machines in this cluster. A machine is considered up-to-date when Machine's UpToDate condition is true.

API Endpoints

The following API endpoints are available:

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusters`

- `DELETE`: delete collection of Cluster
- `GET`: list objects of kind Cluster
- `POST`: create a new Cluster

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusters/{name}`
  - `DELETE`: delete the specified Cluster
  - `GET`: read the specified Cluster
  - `PATCH`: partially update the specified Cluster
  - `PUT`: replace the specified Cluster
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusters/{name}/status`
  - `GET`: read status of the specified Cluster
  - `PATCH`: partially update status of the specified Cluster
  - `PUT`: replace status of the specified Cluster

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusters

HTTP method

`DELETE`

Description

delete collection of Cluster

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind Cluster

HTTP responses

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

HTTP method

`POST`

Description

create a new Cluster

Query parameters

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

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Cluster` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusters/{name}

HTTP method

`DELETE`

Description

delete the specified Cluster

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 Cluster

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified Cluster

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified Cluster

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`Cluster` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Cluster` schema

HTTP code	Response body
201 - Created	`Cluster` schema
401 - Unauthorized	Empty

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusters/{name}/status

HTTP method

`GET`

Description

read status of the specified Cluster

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified Cluster

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified Cluster

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`Cluster` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Cluster` schema
201 - Created	`Cluster` schema

HTTP code	Response body
401 - Unauthorized	Empty

ClusterClass [cluster.x-k8s.io/v1beta1]

Description

ClusterClass is a template which can be used to create managed topologies.

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of ClusterClass.
`status`	`object`	status is the observed state of ClusterClass.

.spec

Description

spec is the desired state of ClusterClass.

Type

`object`

Property	Type	Description
`availabilityGates`	`array`	availabilityGates specifies additional conditions to include when evaluating Cluster Available condition.

Property	Type	Description
		NOTE: this field is considered only for computing v1beta2 conditions. NOTE: If a Cluster is using this ClusterClass, and this Cluster defines a custom list of availabilityGates, such list overrides availabilityGates defined in this field.
`controlPlane`	`object`	controlPlane is a reference to a local struct that holds the details for provisioning the Control Plane for the Cluster.
`infrastructure`	`object`	infrastructure is a reference to a provider-specific template that holds the details for provisioning infrastructure specific cluster for the underlying provider. The underlying provider is responsible for the implementation of the template to an infrastructure cluster.
`infrastructureNamingStrategy`	`object`	infrastructureNamingStrategy allows changing the naming pattern used when creating the infrastructure object.

Property	Type	Description
`patches`	`array`	patches defines the patches which are applied to customize referenced templates of a ClusterClass. Note: Patches will be applied in the order of the array.
`variables`	`array`	variables defines the variables which can be configured in the Cluster topology and are then used in patches.
`workers`	`object`	workers describes the worker nodes for the cluster. It is a collection of node types which can be used to create the worker nodes of the cluster.

.spec.availabilityGates

Description

availabilityGates specifies additional conditions to include when evaluating Cluster Available condition. NOTE: this field is considered only for computing v1beta2 conditions. NOTE: If a Cluster is using this ClusterClass, and this Cluster defines a custom list of availabilityGates, such list overrides availabilityGates defined in this field.

Type

`array`

.spec.availabilityGates[]

Description

ClusterAvailabilityGate contains the type of a Cluster condition to be used as availability gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Cluster's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as availability gates.
`polarity`	`string`	polarity of the conditionType specified in this availabilityGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.controlPlane

Description

controlPlane is a reference to a local struct that holds the details for provisioning the Control Plane for the Cluster.

Type

`object`

Required

`ref`

Property	Type	Description
`machineHealthCheck`	`object`	machineHealthCheck defines a MachineHealthCheck for this ControlPlaneClass. This field is supported if and only if the ControlPlane provider template referenced above is Machine based and supports setting replicas.
`machineInfrastructure`	`object`	machineInfrastructure defines the metadata and infrastructure information for control plane machines. This field is supported if and only if the control plane provider template referenced above is Machine based and supports setting replicas.
`metadata`	`ObjectMeta`	metadata is the metadata applied to the ControlPlane and the Machines of the ControlPlane if the ControlPlaneTemplate referenced is machine based. If not, it is applied only to the ControlPlane. At runtime this metadata is merged with the corresponding metadata from the topology. This field is supported if and only if the control plane provider template referenced is Machine based.

Property	Type	Description
`namingStrategy`	`object`	namingStrategy allows changing the naming pattern used when creating the control plane provider object.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds. NOTE: This value can be overridden while defining a Cluster.Topology.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout` NOTE: This value can be overridden while defining a Cluster.Topology.
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations. NOTE: This value can be overridden while defining a Cluster.Topology.

Property	Type	Description
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: If a Cluster defines a custom list of readinessGates for the control plane, such list overrides readinessGates defined in this field. NOTE: Specific control plane provider implementations might automatically extend the list of readinessGates; e.g. the kubeadm control provider adds ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc.
`ref`	`object`	ref is a required reference to a custom resource offered by a provider.

.spec.controlPlane.machineHealthCheck

Description

machineHealthCheck defines a MachineHealthCheck for this ControlPlaneClass. This field is supported if and only if the ControlPlane provider template referenced above is Machine based and supports setting replicas.

Type

`object`

Property	Type	Description
`maxUnhealthy`	`integer|string`	maxUnhealthy specifies the maximum number of unhealthy machines allowed. Any further remediation is only allowed if at most "maxUnhealthy" machines selected by "selector" are not healthy.
`nodeStartupTimeout`	`string`	nodeStartupTimeout allows to set the maximum time for MachineHealthCheck to consider a Machine unhealthy if a corresponding Node isn't associated through a `Spec.ProviderID` field. The duration set in this field is compared to the greatest of: • Cluster's infrastructure ready condition timestamp (if and when available) • Control Plane's initialized condition timestamp (if and when available) • Machine's infrastructure ready condition timestamp (if and when available) • Machine's metadata creation timestamp Defaults to 10 minutes. If you wish to disable this feature, set the value explicitly to 0.

Property	Type	Description
`remediationTemplate`	`object`	remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.
`unhealthyConditions`	`array`	unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.
`unhealthyRange`	`string`	unhealthyRange specifies the range of unhealthy machines allowed. Any further remediation is only allowed if the number of machines selected by "selector" as not healthy is within the range of "unhealthyRange". Takes precedence over maxUnhealthy. Eg. "[3-5]" - This means that remediation will be allowed only when: (a) there are at least 3 unhealthy machines (and) (b) there are at most 5 unhealthy machines

.spec.controlPlane.machineHealthCheck.remediationTemplate

Description

remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.controlPlane.machineHealthCheck.unhealthyConditions

Description

unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.

Type

`array`

.spec.controlPlane.machineHealthCheck.unhealthyConditions[]

Description

UnhealthyCondition represents a Node condition type and value with a timeout specified as a duration. When the named condition has been in the given status for at least the timeout value, a node is considered unhealthy.

Type

`object`

Required

`status` `timeout` `type`

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`timeout`	`string`	timeout is the duration that a node must be in a given status for, after which the node is considered unhealthy. For example, with a value of "1h", the node must match the status for at least 1 hour before being considered unhealthy.
`type`	`string`	type of Node condition

.spec.controlPlane.machineInfrastructure

Description

machineInfrastructure defines the metadata and infrastructure information for control plane machines. This field is supported if and only if the control plane provider template referenced above is Machine based and supports setting replicas.

Type

`object`

Required

`ref`

Property	Type	Description
`ref`	`object`	ref is a required reference to a custom resource offered by a provider.

.spec.controlPlane.machineInfrastructure.ref

Description

ref is a required reference to a custom resource offered by a provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2

Property	Type	Description
		in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.controlPlane.namingStrategy

Description

namingStrategy allows changing the naming pattern used when creating the control plane provider object.

Type

`object`

Property	Type	Description
`template`	`string`	template defines the template to use for generating the name of the ControlPlane object. If not defined, it will fallback to `{{ .cluster.name }}-{{ .random }}`. If the templated string exceeds 63 characters, it will be trimmed to 58 characters and will get concatenated with a random suffix of length 5. The templating mechanism provides the following arguments: • `.cluster.name`: The name of the cluster object. • `.random`: A random alphanumeric string, without vowels, of length 5.

.spec.controlPlane.readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: If a Cluster defines a custom list of readinessGates for the control plane, such list overrides readinessGates defined in this field. NOTE: Specific control plane provider implementations might automatically extend the list of readinessGates; e.g. the kubeadm control provider adds ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc.

Type

`array`

.spec.controlPlane.readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.controlPlane.ref

Description

ref is a required reference to a custom resource offered by a provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-

Property	Type	Description
		with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.infrastructure

Description

infrastructure is a reference to a provider-specific template that holds the details for provisioning infrastructure specific cluster for the underlying provider. The underlying provider is responsible for the implementation of the template to an infrastructure cluster.

Type

`object`

Required

`ref`

Property	Type	Description
`ref`	`object`	ref is a required reference to a custom resource offered by a provider.

.spec.infrastructure.ref

Description

ref is a required reference to a custom resource offered by a provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

Property	Type	Description
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.infrastructureNamingStrategy

Description

infrastructureNamingStrategy allows changing the naming pattern used when creating the infrastructure object.

Type

`object`

Property	Type	Description
`template`	`string`	template defines the template to use for generating the name of the Infrastructure object. If not defined, it will fallback to `{{ .cluster.name }}-{{ .random }}`. If the templated string exceeds 63 characters, it will be trimmed

Property	Type	Description
		to 58 characters and will get concatenated with a random suffix of length 5. The templating mechanism provides the following arguments: • `.cluster.name`: The name of the cluster object. • `.random`: A random alphanumeric string, without vowels, of length 5.

.spec.patches

Description

patches defines the patches which are applied to customize referenced templates of a ClusterClass. Note: Patches will be applied in the order of the array.

Type

`array`

.spec.patches[]

Description

ClusterClassPatch defines a patch which is applied to customize the referenced templates.

Type

`object`

Required

`name`

Property	Type	Description
`definitions`	`array`	definitions define inline patches. Note: Patches will be applied in the order of the array. Note: Exactly one of Definitions or External must be set.
`description`	`string`	description is a human-readable description of this patch.
`enabledIf`	`string`	enabledIf is a Go template to be used to calculate if a patch should be enabled. It can reference variables defined in .spec.variables and builtin variables. The patch will be enabled if the template evaluates to `true`, otherwise it will be disabled. If EnabledIf is not set, the patch will be enabled per default.
`external`	`object`	external defines an external patch. Note: Exactly one of Definitions or External must be set.
`name`	`string`	name of the patch.

.spec.patches[].definitions

Description

definitions define inline patches. Note: Patches will be applied in the order of the array. Note: Exactly one of Definitions or External must be set.

Type

`array`

.spec.patches[].definitions[]

Description

PatchDefinition defines a patch which is applied to customize the referenced templates.

Type

`object`

Required

`jsonPatches` `selector`

Property	Type	Description
`jsonPatches`	`array`	jsonPatches defines the patches which should be applied on the templates matching the selector. Note: Patches will be applied in the order of the array.
`selector`	`object`	selector defines on which templates the patch should be applied.

.spec.patches[].definitions[].jsonPatches

Description

jsonPatches defines the patches which should be applied on the templates matching the selector. Note: Patches will be applied in the order of the array.

Type

`array`

.spec.patches[].definitions[].jsonPatches[]

Description

JSONPatch defines a JSON patch.

Type

`object`

Required

`op` `path`

Property	Type	Description
`op`	`string`	op defines the operation of the patch. Note: Only `add`, `repla` supported.
`path`	`string`	path defines the path of the patch. Note: Only the spec of a tem the path has to start with /spec/. Note: For now the only allowed `append` and `prepend`, i.e.: • for op: `add`: only index 0 (prepend) and - (append) are allo • for op: `replace` or `remove`: no indexes are allowed
`value`		value defines the value of the patch. Note: Either Value or Value replace operations. Only one of them is allowed to be set at the to use apiextensionsv1.JSON instead of our JSON type, becaus hard-coded schema for apiextensionsv1.JSON which cannot be (unset type field). Ref: https://github.com/kubernetes-sigs/contro tools/blob/d0e03a142d0ecdd5491593e941ee1d6b5d91dba6/pk L111 ↗
`valueFrom`	`object`	valueFrom defines the value of the patch. Note: Either Value or add and replace operations. Only one of them is allowed to be s

.spec.patches[].definitions[].jsonPatches[].valueFrom

Description

valueFrom defines the value of the patch. Note: Either Value or ValueFrom is required for add and replace operations. Only one of them is allowed to be set at the same time.

Type

`object`

Property	Type	Description
`template`	`string`	template is the Go template to be used to calculate the value. A template can reference variables defined in .spec.variables and builtin variables. Note: The template must evaluate to a valid YAML or JSON value.
`variable`	`string`	variable is the variable to be used as value. Variable can be one of the variables defined in .spec.variables or a builtin variable.

.spec.patches[].definitions[].selector

Description

selector defines on which templates the patch should be applied.

Type

`object`

Required

`apiVersion` `kind` `matchResources`

Property	Type	Description
`apiVersion`	`string`	apiVersion filters templates by apiVersion.

Property	Type	Description
`kind`	`string`	kind filters templates by kind.
`matchResources`	`object`	matchResources selects templates based on where they are referenced.

.spec.patches[].definitions[].selector.matchResources

Description

matchResources selects templates based on where they are referenced.

Type

`object`

Property	Type	Description
`controlPlane`	`boolean`	controlPlane selects templates referenced in .spec.ControlPlane. Note: this will match the controlPlane and also the controlPlane machineInfrastructure (depending on the kind and apiVersion).
`infrastructureCluster`	`boolean`	infrastructureCluster selects templates referenced in .spec.infrastructure.

Property	Type	Description
`machineDeploymentClass`	`object`	machineDeploymentClass selects templates referenced in specific MachineDeploymentClasses in .spec.workers.machineDeployments.
`machinePoolClass`	`object`	machinePoolClass selects templates referenced in specific MachinePoolClasses in .spec.workers.machinePools.

.spec.patches[].definitions[].selector.matchResources.machineDeploymentClass

Description

machineDeploymentClass selects templates referenced in specific MachineDeploymentClasses in .spec.workers.machineDeployments.

Type

`object`

Property	Type	Description
`names`	`array`	names selects templates by class names.

.spec.patches[].definitions[].selector.matchResources.machineDeploymentClass.names

Description

names selects templates by class names.

Type

`array`

.spec.patches[].definitions[].selector.matchResources.machineDeploymentClass.names[]

Type

`string`

.spec.patches[].definitions[].selector.matchResources.machinePoolClass

Description

machinePoolClass selects templates referenced in specific MachinePoolClasses in .spec.workers.machinePools.

Type

`object`

Property	Type	Description
`names`	`array`	names selects templates by class names.

.spec.patches[].definitions[].selector.matchResources.machinePoolClass.names

Description

names selects templates by class names.

Type

`array`

.spec.patches[].definitions[].selector.matchResources.machinePoolClass.names[]

Type

`string`

.spec.patches[].external

Description

external defines an external patch. Note: Exactly one of Definitions or External must be set.

Type

`object`

Property	Type	Description
`discoverVariablesExtension`	`string`	discoverVariablesExtension references an extension which is called to discover variables.
`generateExtension`	`string`	generateExtension references an extension which is called to generate patches.
`settings`	`object`	settings defines key value pairs to be passed to the extensions. Values defined here take precedence over the values defined in the corresponding ExtensionConfig.

Property	Type	Description
`validateExtension`	`string`	validateExtension references an extension which is called to validate the topology.

.spec.patches[].external.settings

Description

settings defines key value pairs to be passed to the extensions. Values defined here take precedence over the values defined in the corresponding ExtensionConfig.

Type

`object`

.spec.variables

Description

variables defines the variables which can be configured in the Cluster topology and are then used in patches.

Type

`array`

.spec.variables[]

Description

ClusterClassVariable defines a variable which can be configured in the Cluster topology and used in patches.

Type

`object`

Required

`name` `required` `schema`

Property	Type	Description
`metadata`	`ObjectMeta`	metadata is the metadata of a variable. It can be used to add additional data for higher level tools to a ClusterClassVariable. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please use XMetadata in JSONSchemaProps instead.
`name`	`string`	name of the variable.
`required`	`boolean`	required specifies if the variable is required. Note: this applies to the variable as a whole and thus the top-level object defined in the schema. If nested fields are required, this will be specified inside the schema.
`schema`	`object`	schema defines the schema of the variable.

.spec.variables[].schema

Description

schema defines the schema of the variable.

Type

`object`

Required

`openAPIV3Schema`

Property	Type	Description
`openAPIV3Schema`	`object`	openAPIV3Schema defines the schema of a variable via OpenAPI v3 schema. The schema is a subset of the schema used in Kubernetes CRDs.

.spec.variables[].schema.openAPIV3Schema

Description

openAPIV3Schema defines the schema of a variable via OpenAPI v3 schema. The schema is a subset of the schema used in Kubernetes CRDs.

Type

`object`

Property	Type	Description
`additionalProperties`		additionalProperties specifies the schema of values ir map (keys are always strings). NOTE: Can only be se type is object. NOTE: AdditionalProperties is mutually exclusive with Properties. NOTE: This field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`allOf`		allOf specifies that the variable must validate against of the subschemas in the array. NOTE: This field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`anyOf`		anyOf specifies that the variable must validate agains one or more of the subschemas in the array. NOTE: T

Property	Type	Description
		field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`default`		default is the default value of the variable. NOTE: Can set for all types.
`description`	`string`	description is a human-readable description of this variable.
`enum`	`array`	enum is the list of valid values of the variable. NOTE: Can be set for all types.
`example`		example is an example for this variable.
`exclusiveMaximum`	`boolean`	exclusiveMaximum specifies if the Maximum is exclus NOTE: Can only be set if type is integer or number.
`exclusiveMinimum`	`boolean`	exclusiveMinimum specifies if the Minimum is exclusi NOTE: Can only be set if type is integer or number.
`format`	`string`	format is an OpenAPI v3 format string. Unknown form are ignored. For a list of supported formats please se (of the k8s.io/apiextensions-apiserver version we're currently using) https://github.com/kubernetes/apiextensions-

Property	Type	Description
		apiserver/blob/master/pkg/apiserver/validation/format ↗ NOTE: Can only be set if type is string.
`items`		items specifies fields of an array. NOTE: Can only be if type is array. NOTE: This field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`maxItems`	`integer`	maxItems is the max length of an array variable. NOT Can only be set if type is array.
`maxLength`	`integer`	maxLength is the max length of a string variable. NO Can only be set if type is string.
`maxProperties`	`integer`	maxProperties is the maximum amount of entries in a map or properties in an object. NOTE: Can only be se type is object.
`maximum`	`integer`	maximum is the maximum of an integer or number variable. If ExclusiveMaximum is false, the variable is valid if it is lower than, or equal to, the value of Maxim If ExclusiveMaximum is true, the variable is valid if it i strictly lower than the value of Maximum. NOTE: Can only be set if type is integer or number.

Property	Type	Description
`minItems`	`integer`	minItems is the min length of an array variable. NOTE Can only be set if type is array.
`minLength`	`integer`	minLength is the min length of a string variable. NOTI Can only be set if type is string.
`minProperties`	`integer`	minProperties is the minimum amount of entries in a r or properties in an object. NOTE: Can only be set if ty is object.
`minimum`	`integer`	minimum is the minimum of an integer or number variable. If ExclusiveMinimum is false, the variable is valid if it is greater than, or equal to, the value of Minimum. If ExclusiveMinimum is true, the variable is valid if it is strictly greater than the value of Minimum. NOTE: Can only be set if type is integer or number.
`not`		not specifies that the variable must not validate again the subschema. NOTE: This field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`oneOf`		oneOf specifies that the variable must validate agains exactly one of the subschemas in the array. NOTE: T field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.

Property	Type	Description
`pattern`	`string`	pattern is the regex which a string variable must matc NOTE: Can only be set if type is string.
`properties`		properties specifies fields of an object. NOTE: Can or be set if type is object. NOTE: Properties is mutually exclusive with AdditionalProperties. NOTE: This field uses PreserveUnknownFields and Schemaless, beca recursive validation is not possible.
`required`	`array`	required specifies which fields of an object are require NOTE: Can only be set if type is object.
`type`	`string`	type is the type of the variable. Valid values are: objec array, string, integer, number or boolean.
`uniqueItems`	`boolean`	uniqueItems specifies if items in an array must be unique. NOTE: Can only be set if type is array.
`x-kubernetes-int-or-string`	`boolean`	x-kubernetes-int-or-string specifies that this value is either an integer or a string. If this is true, an empty ty is allowed and type as child of anyOf is permitted if following one of the following patterns: 1. anyOf: • type: integer • type: string

Property	Type	Description

		2. allOf:
		• anyOf:
		  • type: integer
		  • type: string
		• ... zero or more

`x-kubernetes-preserve-unknown-fields`	`boolean`	x-kubernetes-preserve-unknown-fields allows setting fields in a variable object which are not defined in the variable schema. This affects fields recursively, excep nested properties or additionalProperties are specifie the schema.
`x-kubernetes-validations`	`array`	x-kubernetes-validations describes a list of validation rules written in the CEL expression language.
`x-metadata`	`object`	x-metadata is the metadata of a variable or a nested f within a variable. It can be used to add additional data higher level tools.

.spec.variables[].schema.openAPIV3Schema.enum

Description

enum is the list of valid values of the variable. NOTE: Can be set for all types.

Type

`array`

.spec.variables[].schema.openAPIV3Schema.enum[]

Type

.spec.variables[].schema.openAPIV3Schema.required

Description

required specifies which fields of an object are required. NOTE: Can only be set if type is object.

Type

`array`

.spec.variables[].schema.openAPIV3Schema.required[]

Type

`string`

.spec.variables[].schema.openAPIV3Schema.x-kubernetes-validations

Description

x-kubernetes-validations describes a list of validation rules written in the CEL expression language.

Type

`array`

.spec.variables[].schema.openAPIV3Schema.x-kubernetes-validations[]

Description

ValidationRule describes a validation rule written in the CEL expression language.

Type

`object`

Required

`rule`

Property	Type	Description
`fieldPath`	`string`	fieldPath represents the field path returned when the validation fails. It must be a relative JSON path (i.e. with array notation) scoped to the location of this x-kubernetes-validations extension in the schema and refer to an existing field. e.g. when validation checks if a specific attribute `foo` under a map `testMap`, the fieldPath could be set to `.testMap.foo` If the validation checks two lists must have unique attributes, the fieldPath could be set to either of the list: e.g. `.testList` It does not support list numeric index. It supports child operation to refer to an existing field currently. Refer to JSONPath support in Kubernetes ↗ for more info. Numeric index of array is not supported. For field name which contains special characters, use `['specialName']` to refer the field name. e.g. for attribute `foo.34$` appears in a list `testList`, the fieldPath could be set to `.testList['foo.34$']`

Property	Type	Description
`message`	`string`	message represents the message displayed when validation fails. The message is required if the Rule contains line breaks. The message must not contain line breaks. If unset, the message is "failed rule: {Rule}". e.g. "must be a URL with the host matching spec.host"
`messageExpression`	`string`	messageExpression declares a CEL expression that evaluates to the validation failure message that is returned when this rule fails. Since messageExpression is used as a failure message, it must evaluate to a string. If both message and messageExpression are present on a rule, then messageExpression will be used if validation fails. If messageExpression results in a runtime error, the validation failure message is produced as if the messageExpression field were unset. If messageExpression evaluates to an empty string, a string with only spaces, or a string that contains line breaks, then the validation failure message will also be produced as if the messageExpression field were unset. messageExpression has access to all the same variables as the rule; the only difference is the return type. Example: "x must be less than max ("+string(self.max)+")"
`reason`	`string`	reason provides a machine-readable validation failure reason that is returned to the caller when a request fails this validation rule. The currently supported reasons are: "FieldValueInvalid", "FieldValueForbidden", "FieldValueRequired",

Property	Type	Description
		"FieldValueDuplicate". If not set, default to use "FieldValueInvalid". All future added reasons must be accepted by clients when reading this value and unknown reasons should be treated as FieldValueInvalid.
`rule`	`string`	rule represents the expression which will be evaluated by CEL. ref: https://github.com/google/cel-spec ↗ The Rule is scoped to the location of the x-kubernetes-validations extension in the schema. The `self` variable in the CEL expression is bound to the scoped value. If the Rule is scoped to an object with properties, the accessible properties of the object are field selectable via `self.field` and field presence can be checked via `has(self.field)`. If the Rule is scoped to an object with additionalProperties (i.e. a map) the value of the map are accessible via `self[mapKey]`, map containment can be checked via `mapKey in self` and all entries of the map are accessible via CEL macros and functions such as `self.all(...)`. If the Rule is scoped to an array, the elements of the array are accessible via `self[i]` and also by macros and functions. If the Rule is scoped to a scalar, `self` is bound to the scalar value. Examples: • Rule scoped to a map of objects: {"rule": "self.components['Widget'].priority < 10"} • Rule scoped to a list of integers: {"rule": "self.values.all(value, value >= 0 && value < 100)"}

Property	Type	Description

		• Rule scoped to a string value: {"rule": "self.startsWith('kube')"}
		
		Unknown data preserved in custom resources via x-kubernetes-preserve-unknown-fields is not accessible in CEL expressions. This includes:
		
		• Unknown field values that are preserved by object schemas with x-kubernetes-preserve-unknown-fields.
		• Object properties where the property schema is of an "unknown type". An "unknown type" is recursively defined as:
		  • A schema with no type and x-kubernetes-preserve-unknown-fields set to true
		  • An array where the items schema is of an "unknown type"
		  • An object where the additionalProperties schema is of an "unknown type"
		
		Only property names of the form `[a-zA-Z_.-/][a-zA-Z0-9_.-/]*` are accessible. Accessible property names are escaped according to the following rules when accessed in the expression:
		
		• '__' escapes to '**underscores**'
		• '.' escapes to '**dot**'
		• '-' escapes to '**dash**'
		• '/' escapes to '**slash**'
		• Property names that exactly match a CEL RESERVED keyword escape to '**{keyword}**'. The keywords are: "true", "false", "null", "in", "as", "break", "const", "continue", "else", "for",

Property	Type	Description
		"function", "if", "import", "let", "loop", "package", "namespace", "return". Examples: • Rule accessing a property named "namespace": {"rule": "self.**namespace** > 0"} • Rule accessing a property named "x-prop": {"rule": "self.x__dash__prop > 0"} • Rule accessing a property named "redact__d": {"rule": "self.redact__underscores__d > 0"} If `rule` makes use of the `oldSelf` variable it is implicitly a `transition rule`. By default, the `oldSelf` variable is the same type as `self`. Transition rules by default are applied only on UPDATE requests and are skipped if an old value could not be found.

.spec.variables[].schema.openAPIV3Schema.x-metadata

Description

x-metadata is the metadata of a variable or a nested field within a variable. It can be used to add additional data for higher level tools.

Type

`object`

Property	Type	Description
`annotations`	`object`	annotations is an unstructured key value map that can be used to store and retrieve arbitrary metadata. They are not

Property	Type	Description
		queryable.
`labels`	`object`	labels is a map of string keys and values that can be used to organize and categorize (scope and select) variables.

.spec.variables[].schema.openAPIV3Schema.x-metadata.annotations

Description

annotations is an unstructured key value map that can be used to store and retrieve arbitrary metadata. They are not queryable.

Type

`object`

.spec.variables[].schema.openAPIV3Schema.x-metadata.labels

Description

labels is a map of string keys and values that can be used to organize and categorize (scope and select) variables.

Type

`object`

.spec.workers

Description

workers describes the worker nodes for the cluster. It is a collection of node types which can be used to create the worker nodes of the cluster.

Type

`object`

Property	Type	Description
`machineDeployments`	`array`	machineDeployments is a list of machine deployment classes that can be used to create a set of worker nodes.
`machinePools`	`array`	machinePools is a list of machine pool classes that can be used to create a set of worker nodes.

.spec.workers.machineDeployments

Description

machineDeployments is a list of machine deployment classes that can be used to create a set of worker nodes.

Type

`array`

.spec.workers.machineDeployments[]

Description

MachineDeploymentClass serves as a template to define a set of worker nodes of the cluster provisioned using the `ClusterClass`.

Type

`object`

Required

`class` `template`

Property	Type	Description
`class`	`string`	class denotes a type of worker node present in the cluster, this name MUST be unique within a ClusterClass and can be referenced in the Cluster to create a managed MachineDeployment.
`failureDomain`	`string`	failureDomain is the failure domain the machines will be created in. Must match a key in the FailureDomains map stored on the cluster object. NOTE: This value can be overridden while defining a Cluster.Topology using this MachineDeploymentClass.
`machineHealthCheck`	`object`	machineHealthCheck defines a MachineHealthCheck for this MachineDeploymentClass.
`minReadySeconds`	`integer`	minReadySeconds is the minimum number of seconds for which a newly created machine should be ready. Defaults to 0 (machine will be considered available as soon as it is ready) NOTE: This value can be overridden while defining a Cluster.Topology using this MachineDeploymentClass.

Property	Type	Description
`namingStrategy`	`object`	namingStrategy allows changing the naming pattern used when creating the MachineDeployment.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds. NOTE: This value can be overridden while defining a Cluster.Topology using this MachineDeploymentClass.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout` NOTE: This value can be overridden while defining a Cluster.Topology using this MachineDeploymentClass.
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations. NOTE: This value can be

Property	Type	Description
		overridden while defining a Cluster.Topology using this MachineDeploymentClass.
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: If a Cluster defines a custom list of readinessGates for a MachineDeployment using this MachineDeploymentClass, such list overrides readinessGates defined in this field.
`strategy`	`object`	strategy is the deployment strategy to use to replace existing machines with new ones. NOTE: This value can be overridden while defining a Cluster.Topology using this MachineDeploymentClass.
`template`	`object`	template is a local struct containing a collection of templates for creation of

Property	Type	Description
		MachineDeployment objects representing a set of worker nodes.

.spec.workers.machineDeployments[].machineHealthCheck

Description

machineHealthCheck defines a MachineHealthCheck for this MachineDeploymentClass.

Type

`object`

Property	Type	Description
`maxUnhealthy`	`integer|string`	maxUnhealthy specifies the maximum number of unhealthy machines allowed. Any further remediation is only allowed if at most "maxUnhealthy" machines selected by "selector" are not healthy.
`nodeStartupTimeout`	`string`	nodeStartupTimeout allows to set the maximum time for MachineHealthCheck to consider a Machine unhealthy if a corresponding Node isn't associated through a `Spec.ProviderID` field. The duration set in this field is compared to the greatest of: • Cluster's infrastructure ready condition timestamp (if and when available)

Property	Type	Description
		• Control Plane's initialized condition timestamp (if and when available) • Machine's infrastructure ready condition timestamp (if and when available) • Machine's metadata creation timestamp Defaults to 10 minutes. If you wish to disable this feature, set the value explicitly to 0.
`remediationTemplate`	`object`	remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.
`unhealthyConditions`	`array`	unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.
`unhealthyRange`	`string`	unhealthyRange specifies the range of unhealthy machines allowed. Any further remediation is only allowed if the number of machines selected by "selector" as not healthy is within the range of "unhealthyRange". Takes precedence over

Property	Type	Description
		maxUnhealthy. Eg. "[3-5]" - This means that remediation will be allowed only when: (a) there are at least 3 unhealthy machines (and) (b) there are at most 5 unhealthy machines

.spec.workers.machineDeployments[].machineHealthCheck.remediationTemplate

Description

remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2

Property	Type	Description
		in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.workers.machineDeployments[].machineHealthCheck.unhealthyConditions

Description

unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.

Type

`array`

.spec.workers.machineDeployments[].machineHealthCheck.unhealthyConditions[]

Description

UnhealthyCondition represents a Node condition type and value with a timeout specified as a duration. When the named condition has been in the given status for at least the timeout value, a node is considered unhealthy.

Type

`object`

Required

`status` `timeout` `type`

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.

Property	Type	Description
`timeout`	`string`	timeout is the duration that a node must be in a given status for, after which the node is considered unhealthy. For example, with a value of "1h", the node must match the status for at least 1 hour before being considered unhealthy.
`type`	`string`	type of Node condition

.spec.workers.machineDeployments[].namingStrategy

Description

namingStrategy allows changing the naming pattern used when creating the MachineDeployment.

Type

`object`

Property	Type	Description
`template`	`string`	template defines the template to use for generating the name of the MachineDeployment object. If not defined, it will fallback to `{{ .cluster.name }}-{{ .machineDeployment.topologyNam }}-{{ .random }}`. If the templated string exceeds 63 characters, it will be trimmed to 58 characters and will get concatenated with a random suffix of length 5. The templating mechanism provides the following arguments: • `.cluster.name`: The name of the cluster object. • `.random`: A random alphanumeric string, without vowels, of length 5.

Property	Type	Description
		• `.machineDeployment.topologyName`: The name of the MachineDeployment topology (Cluster.spec.topology.workers.machineDeployments[].name

.spec.workers.machineDeployments[].readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: If a Cluster defines a custom list of readinessGates for a MachineDeployment using this MachineDeploymentClass, such list overrides readinessGates defined in this field.

Type

`array`

.spec.workers.machineDeployments[].readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't

Property	Type	Description
		exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.workers.machineDeployments[].strategy

Description

strategy is the deployment strategy to use to replace existing machines with new ones. NOTE: This value can be overridden while defining a Cluster.Topology using this MachineDeploymentClass.

Type

`object`

Property	Type	Description
`remediation`	`object`	remediation controls the strategy of remediating unhealthy machines and how remediating operations should occur during the lifecycle of the dependant MachineSets.
`rollingUpdate`	`object`	rollingUpdate is the rolling update config params. Present only if MachineDeploymentStrategyType =

Property	Type	Description
		RollingUpdate.
`type`	`string`	type of deployment. Allowed values are RollingUpdate and OnDelete. The default is RollingUpdate.

.spec.workers.machineDeployments[].strategy.remediation

Description

remediation controls the strategy of remediating unhealthy machines and how remediating operations should occur during the lifecycle of the dependant MachineSets.

Type

`object`

Property	Type	Description
`maxInFlight`	`integer|string`	maxInFlight determines how many in flight remediations should happen at the same time. Remediation only happens on the MachineSet with the most current revision, while older MachineSets (usually present during rollout operations) aren't allowed to remediate. Note: In general (independent of remediations), unhealthy machines are always prioritized during scale down operations over healthy ones. MaxInFlight can be set to a fixed number or a percentage. Example: when this is set to 20%, the

Property	Type	Description
		MachineSet controller deletes at most 20% of the desired replicas. If not set, remediation is limited to all machines (bounded by replicas) under the active MachineSet's management.

.spec.workers.machineDeployments[].strategy.rollingUpdate

Description

rollingUpdate is the rolling update config params. Present only if MachineDeploymentStrategyType = RollingUpdate.

Type

`object`

Property	Type	Description
`deletePolicy`	`string`	deletePolicy defines the policy used by the MachineDeployment to identify nodes to delete when downscaling. Valid values are "Random", "Newest", "Oldest" When no value is supplied, the default DeletePolicy of MachineSet is used
`maxSurge`	`integer|string`	maxSurge is the maximum number of machines that can be scheduled above the desired number of machines. Value can be an absolute number (ex: 5) or a percentage of desired machines (ex: 10%). This can not be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up. Defaults to 1. Example: when this is set to 30%,

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

.spec.workers.machineDeployments[].template

Description

template is a local struct containing a collection of templates for creation of MachineDeployment objects representing a set of worker nodes.

Type

`object`

Required

`bootstrap` `infrastructure`

Property	Type	Description
`bootstrap`	`object`	bootstrap contains the bootstrap template reference to be used for the creation of worker Machines.
`infrastructure`	`object`	infrastructure contains the infrastructure template reference to be used for the creation of worker Machines.
`metadata`	`ObjectMeta`	metadata is the metadata applied to the MachineDeployment and the machines of the MachineDeployment. At runtime this metadata is merged with the corresponding metadata from the topology.

.spec.workers.machineDeployments[].template.bootstrap

Description

bootstrap contains the bootstrap template reference to be used for the creation of worker Machines.

Type

`object`

Required

`ref`

Property	Type	Description
`ref`	`object`	ref is a required reference to a custom resource offered by a provider.

.spec.workers.machineDeployments[].template.bootstrap.ref

Description

ref is a required reference to a custom resource offered by a provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

Property	Type	Description
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.workers.machineDeployments[].template.infrastructure

Description

infrastructure contains the infrastructure template reference to be used for the creation of worker Machines.

Type

`object`

Required

`ref`

Property	Type	Description
`ref`	`object`	ref is a required reference to a custom resource offered by a provider.

.spec.workers.machineDeployments[].template.infrastructure.ref

Description

ref is a required reference to a custom resource offered by a provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}"

Property	Type	Description
		(where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.workers.machinePools

Description

machinePools is a list of machine pool classes that can be used to create a set of worker nodes.

Type

`array`

.spec.workers.machinePools[]

Description

MachinePoolClass serves as a template to define a pool of worker nodes of the cluster provisioned using `ClusterClass`.

Type

`object`

Required

`class` `template`

Property	Type	Description
`class`	`string`	class denotes a type of machine pool present in the cluster, this name MUST be unique within a ClusterClass and can be referenced in the Cluster to create a managed MachinePool.
`failureDomains`	`array`	failureDomains is the list of failure domains the MachinePool should be attached to. Must match a key in the FailureDomains map stored on the cluster object. NOTE: This value can be overridden while defining a

Property	Type	Description
		Cluster.Topology using this MachinePoolClass.
`minReadySeconds`	`integer`	minReadySeconds is the minimum number of seconds for which a newly created machine pool should be ready. Defaults to 0 (machine will be considered available as soon as it is ready) NOTE: This value can be overridden while defining a Cluster.Topology using this MachinePoolClass.
`namingStrategy`	`object`	namingStrategy allows changing the naming pattern used when creating the MachinePool.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine Pool is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds. NOTE: This value can be overridden while defining a Cluster.Topology using this MachinePoolClass.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`

Property	Type	Description
		NOTE: This value can be overridden while defining a Cluster.Topology using this MachinePoolClass.
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations. NOTE: This value can be overridden while defining a Cluster.Topology using this MachinePoolClass.
`template`	`object`	template is a local struct containing a collection of templates for creation of MachinePools objects representing a pool of worker nodes.

.spec.workers.machinePools[].failureDomains

Description

failureDomains is the list of failure domains the MachinePool should be attached to. Must match a key in the FailureDomains map stored on the cluster object. NOTE: This value can be overridden while defining a Cluster.Topology using this MachinePoolClass.

Type

`array`

.spec.workers.machinePools[].failureDomains[]

Type

`string`

.spec.workers.machinePools[].namingStrategy

Description

namingStrategy allows changing the naming pattern used when creating the MachinePool.

Type

`object`

Property	Type	Description
`template`	`string`	template defines the template to use for generating the name of the MachinePool object. If not defined, it will fallback to `{{ .cluster.name }}-{{ .machinePool.topologyName }}-{{ .random }}`. If the templated string exceeds 63 characters, it will be trimmed to 58 characters and will get concatenated with a random suffix of length 5. The templating mechanism provides the following arguments: • `.cluster.name`: The name of the cluster object. • `.random`: A random alphanumeric string, without vowels, of length 5. • `.machinePool.topologyName`: The name of the MachinePool topology (Cluster.spec.topology.workers.machinePools[].name).

.spec.workers.machinePools[].template

Description

template is a local struct containing a collection of templates for creation of MachinePools objects representing a pool of worker nodes.

Type

`object`

Required

`bootstrap` `infrastructure`

Property	Type	Description
`bootstrap`	`object`	bootstrap contains the bootstrap template reference to be used for the creation of the Machines in the MachinePool.
`infrastructure`	`object`	infrastructure contains the infrastructure template reference to be used for the creation of the MachinePool.
`metadata`	`ObjectMeta`	metadata is the metadata applied to the MachinePool. At runtime this metadata is merged with the corresponding metadata from the topology.

.spec.workers.machinePools[].template.bootstrap

Description

bootstrap contains the bootstrap template reference to be used for the creation of the Machines in the MachinePool.

Type

`object`

Required

`ref`

Property	Type	Description
`ref`	`object`	ref is a required reference to a custom resource offered by a provider.

.spec.workers.machinePools[].template.bootstrap.ref

Description

ref is a required reference to a custom resource offered by a provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-

Property	Type	Description
		architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.workers.machinePools[].template.infrastructure

Description

infrastructure contains the infrastructure template reference to be used for the creation of the MachinePool.

Type

`object`

Required

`ref`

Property	Type	Description
`ref`	`object`	ref is a required reference to a custom resource offered by a provider.

.spec.workers.machinePools[].template.infrastructure.ref

Description

ref is a required reference to a custom resource offered by a provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

Property	Type	Description
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.status

Description

status is the observed state of ClusterClass.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions defines current observed state of the ClusterClass.
`observedGeneration`	`integer`	observedGeneration is the latest generation observed by the controller.
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in ClusterClass's status with the V1Beta2 version.
`variables`	`array`	variables is a list of ClusterClassStatusVariable that are defined for the ClusterClass.

.status.conditions

Description

conditions defines current observed state of the ClusterClass.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in ClusterClass's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions represents the observations of a ClusterClass's current state. Known condition types are VariablesReady, RefVersionsUpToDate, Paused.

.status.v1beta2.conditions

Description

conditions represents the observations of a ClusterClass's current state. Known condition types are VariablesReady, RefVersionsUpToDate, Paused.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the

Property	Type	Description
		condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

.status.variables

Description

variables is a list of ClusterClassStatusVariable that are defined for the ClusterClass.

Type

`array`

.status.variables[]

Description

ClusterClassStatusVariable defines a variable which appears in the status of a ClusterClass.

Type

`object`

Required

`definitions` `name`

Property	Type	Description
`definitions`	`array`	definitions is a list of definitions for a variable.
`definitionsConflict`	`boolean`	definitionsConflict specifies whether or not there are conflicting definitions for a single variable name.
`name`	`string`	name is the name of the variable.

.status.variables[].definitions

Description

definitions is a list of definitions for a variable.

Type

`array`

.status.variables[].definitions[]

Description

ClusterClassStatusVariableDefinition defines a variable which appears in the status of a ClusterClass.

Type

`object`

Required

`from` `required` `schema`

Property	Type	Description
`from`	`string`	from specifies the origin of the variable definition. This will be `inline` for variables defined in the ClusterClass or the name of a patch defined in the ClusterClass for variables discovered from a DiscoverVariables runtime extensions.
`metadata`	`ObjectMeta`	metadata is the metadata of a variable. It can be used to add additional data for higher level tools to a ClusterClassVariable. Deprecated: This field is deprecated and is going to be removed in the next apiVersion.
`required`	`boolean`	required specifies if the variable is required. Note: this applies to the variable as a whole and thus the top-level object defined in the schema. If nested fields are required, this will be specified inside the schema.
`schema`	`object`	schema defines the schema of the variable.

.status.variables[].definitions[].schema

Description

schema defines the schema of the variable.

Type

`object`

Required

`openAPIV3Schema`

Property	Type	Description
`openAPIV3Schema`	`object`	openAPIV3Schema defines the schema of a variable via OpenAPI v3 schema. The schema is a subset of the schema used in Kubernetes CRDs.

.status.variables[].definitions[].schema.openAPIV3Schema

Description

openAPIV3Schema defines the schema of a variable via OpenAPI v3 schema. The schema is a subset of the schema used in Kubernetes CRDs.

Type

`object`

Property	Type	Description
`additionalProperties`		additionalProperties specifies the schema of values ir map (keys are always strings). NOTE: Can only be se type is object. NOTE: AdditionalProperties is mutually exclusive with Properties. NOTE: This field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.

Property	Type	Description
`allOf`		allOf specifies that the variable must validate against of the subschemas in the array. NOTE: This field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`anyOf`		anyOf specifies that the variable must validate against one or more of the subschemas in the array. NOTE: This field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`default`		default is the default value of the variable. NOTE: Can be set for all types.
`description`	`string`	description is a human-readable description of this variable.
`enum`	`array`	enum is the list of valid values of the variable. NOTE: Can be set for all types.
`example`		example is an example for this variable.
`exclusiveMaximum`	`boolean`	exclusiveMaximum specifies if the Maximum is exclusive. NOTE: Can only be set if type is integer or number.

Property	Type	Description
`exclusiveMinimum`	`boolean`	exclusiveMinimum specifies if the Minimum is exclusi NOTE: Can only be set if type is integer or number.
`format`	`string`	format is an OpenAPI v3 format string. Unknown form are ignored. For a list of supported formats please se (of the k8s.io/apiextensions-apiserver version we're currently using) https://github.com/kubernetes/apiextensions-apiserver/blob/master/pkg/apiserver/validation/format ↗ NOTE: Can only be set if type is string.
`items`		items specifies fields of an array. NOTE: Can only be if type is array. NOTE: This field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`maxItems`	`integer`	maxItems is the max length of an array variable. NOT Can only be set if type is array.
`maxLength`	`integer`	maxLength is the max length of a string variable. NO Can only be set if type is string.
`maxProperties`	`integer`	maxProperties is the maximum amount of entries in a map or properties in an object. NOTE: Can only be se type is object.

Property	Type	Description
`maximum`	`integer`	maximum is the maximum of an integer or number variable. If ExclusiveMaximum is false, the variable is valid if it is lower than, or equal to, the value of Maxim If ExclusiveMaximum is true, the variable is valid if it i strictly lower than the value of Maximum. NOTE: Can only be set if type is integer or number.
`minItems`	`integer`	minItems is the min length of an array variable. NOTE Can only be set if type is array.
`minLength`	`integer`	minLength is the min length of a string variable. NOTI Can only be set if type is string.
`minProperties`	`integer`	minProperties is the minimum amount of entries in a r or properties in an object. NOTE: Can only be set if ty is object.
`minimum`	`integer`	minimum is the minimum of an integer or number variable. If ExclusiveMinimum is false, the variable is valid if it is greater than, or equal to, the value of Minimum. If ExclusiveMinimum is true, the variable is valid if it is strictly greater than the value of Minimum. NOTE: Can only be set if type is integer or number.
`not`		not specifies that the variable must not validate again the subschema. NOTE: This field uses

Property	Type	Description
		PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`oneOf`		oneOf specifies that the variable must validate agains exactly one of the subschemas in the array. NOTE: T field uses PreserveUnknownFields and Schemaless, because recursive validation is not possible.
`pattern`	`string`	pattern is the regex which a string variable must matc NOTE: Can only be set if type is string.
`properties`		properties specifies fields of an object. NOTE: Can or be set if type is object. NOTE: Properties is mutually exclusive with AdditionalProperties. NOTE: This field uses PreserveUnknownFields and Schemaless, beca recursive validation is not possible.
`required`	`array`	required specifies which fields of an object are require NOTE: Can only be set if type is object.
`type`	`string`	type is the type of the variable. Valid values are: objec array, string, integer, number or boolean.
`uniqueItems`	`boolean`	uniqueItems specifies if items in an array must be unique. NOTE: Can only be set if type is array.

Property	Type	Description
`x-kubernetes-int-or-string`	`boolean`	x-kubernetes-int-or-string specifies that this value is either an integer or a string. If this is true, an empty ty is allowed and type as child of anyOf is permitted if following one of the following patterns: 1. anyOf: • type: integer • type: string 2. allOf: • anyOf: • type: integer • type: string • ... zero or more
`x-kubernetes-preserve-unknown-fields`	`boolean`	x-kubernetes-preserve-unknown-fields allows setting fields in a variable object which are not defined in the variable schema. This affects fields recursively, excep nested properties or additionalProperties are specifie the schema.
`x-kubernetes-validations`	`array`	x-kubernetes-validations describes a list of validation rules written in the CEL expression language.
`x-metadata`	`object`	x-metadata is the metadata of a variable or a nested f within a variable. It can be used to add additional data higher level tools.

.status.variables[].definitions[].schema.openAPIV3Schema.enum

Description

enum is the list of valid values of the variable. NOTE: Can be set for all types.

Type

`array`

.status.variables[].definitions[].schema.openAPIV3Schema.enum[]

Type

.status.variables[].definitions[].schema.openAPIV3Schema.required

Description

required specifies which fields of an object are required. NOTE: Can only be set if type is object.

Type

`array`

.status.variables[].definitions[].schema.openAPIV3Schema.required[]

Type

`string`

.status.variables[].definitions[].schema.openAPIV3Schema.x-kubernetes-validations

Description

x-kubernetes-validations describes a list of validation rules written in the CEL expression language.

Type

`array`

.status.variables[].definitions[].schema.openAPIV3Schema.x-kubernetes-validations[]

Description

ValidationRule describes a validation rule written in the CEL expression language.

Type

`object`

Required

`rule`

Property	Type	Description
`fieldPath`	`string`	fieldPath represents the field path returned when the validation fails. It must be a relative JSON path (i.e. with array notation) scoped to the location of this x-kubernetes-validations extension in the schema and refer to an existing field. e.g. when validation checks if a specific attribute `foo` under a map `testMap`, the fieldPath could be set to `.testMap.foo` If the validation checks two lists must have unique attributes, the fieldPath could be set to either of the list: e.g. `.testList` It does not support list numeric index. It supports child operation to refer to an existing field currently. Refer to JSONPath support in Kubernetes ↗ for more info. Numeric index of array is not supported. For field name which contains special characters, use

Property	Type	Description
		`['specialName']` to refer the field name. e.g. for attribute `foo.34$` appears in a list `testList`, the fieldPath could be set to `.testList['foo.34$']`
`message`	`string`	message represents the message displayed when validation fails. The message is required if the Rule contains line breaks. The message must not contain line breaks. If unset, the message is "failed rule: {Rule}". e.g. "must be a URL with the host matching spec.host"
`messageExpression`	`string`	messageExpression declares a CEL expression that evaluates to the validation failure message that is returned when this rule fails. Since messageExpression is used as a failure message, it must evaluate to a string. If both message and messageExpression are present on a rule, then messageExpression will be used if validation fails. If messageExpression results in a runtime error, the validation failure message is produced as if the messageExpression field were unset. If messageExpression evaluates to an empty string, a string with only spaces, or a string that contains line breaks, then the validation failure message will also be produced as if the messageExpression field were unset. messageExpression has access to all the same variables as the rule; the only difference is the return type. Example: "x must be less than max ("+string(self.max)+")"

Property	Type	Description
`reason`	`string`	reason provides a machine-readable validation failure reason that is returned to the caller when a request fails this validation rule. The currently supported reasons are: "FieldValueInvalid", "FieldValueForbidden", "FieldValueRequired", "FieldValueDuplicate". If not set, default to use "FieldValueInvalid". All future added reasons must be accepted by clients when reading this value and unknown reasons should be treated as FieldValueInvalid.
`rule`	`string`	rule represents the expression which will be evaluated by CEL. ref: https://github.com/google/cel-spec ↗ The Rule is scoped to the location of the x-kubernetes-validations extension in the schema. The `self` variable in the CEL expression is bound to the scoped value. If the Rule is scoped to an object with properties, the accessible properties of the object are field selectable via `self.field` and field presence can be checked via `has(self.field)`. If the Rule is scoped to an object with additionalProperties (i.e. a map) the value of the map are accessible via `self[mapKey]`, map containment can be checked via `mapKey in self` and all entries of the map are accessible via CEL macros and functions such as `self.all(...)`. If the Rule is scoped to an array, the elements of the array are accessible via `self[i]` and also by macros and functions. If the Rule is scoped to a scalar, `self` is bound to the scalar value. Examples:

Property	Type	Description

		• Rule scoped to a map of objects: {"rule": "self.components['Widget'].priority < 10"}
		• Rule scoped to a list of integers: {"rule": "self.values.all(value, value >= 0 && value < 100)"}
		• Rule scoped to a string value: {"rule": "self.startsWith('kube')"}
		
		Unknown data preserved in custom resources via x-kubernetes-preserve-unknown-fields is not accessible in CEL expressions. This includes:
		
		• Unknown field values that are preserved by object schemas with x-kubernetes-preserve-unknown-fields.
		• Object properties where the property schema is of an "unknown type". An "unknown type" is recursively defined as:
		  • A schema with no type and x-kubernetes-preserve-unknown-fields set to true
		  • An array where the items schema is of an "unknown type"
		  • An object where the additionalProperties schema is of an "unknown type"
		
		Only property names of the form `[a-zA-Z_.-/][a-zA-Z0-9_.-/]*` are accessible. Accessible property names are escaped according to the following rules when accessed in the expression:
		
		• '__' escapes to '**underscores**'
		• '.' escapes to '**dot**'
		• '-' escapes to '**dash**'

Property	Type	Description

		• '/' escapes to '**slash**'
		• Property names that exactly match a CEL RESERVED keyword escape to '**{keyword}**'. The keywords are: "true", "false", "null", "in", "as", "break", "const", "continue", "else", "for", "function", "if", "import", "let", "loop", "package", "namespace", "return". Examples:
		  • Rule accessing a property named "namespace": {"rule": "self.**namespace** > 0"}
		  • Rule accessing a property named "x-prop": {"rule": "self.x__dash__prop > 0"}
		  • Rule accessing a property named "redact__d": {"rule": "self.redact__underscores__d > 0"}
		If `rule` makes use of the `oldSelf` variable it is implicitly a `transition rule`.
		By default, the `oldSelf` variable is the same type as `self`.
		Transition rules by default are applied only on UPDATE requests and are skipped if an old value could not be found.

.status.variables[].definitions[].schema.openAPIV3Schema.x-metadata

Description

x-metadata is the metadata of a variable or a nested field within a variable. It can be used to add additional data for higher level tools.

Type

`object`

Property	Type	Description
`annotations`	`object`	annotations is an unstructured key value map that can be used to store and retrieve arbitrary metadata. They are not queryable.
`labels`	`object`	labels is a map of string keys and values that can be used to organize and categorize (scope and select) variables.

.status.variables[].definitions[].schema.openAPIV3Schema.x-metadata.annotations

Description

annotations is an unstructured key value map that can be used to store and retrieve arbitrary metadata. They are not queryable.

Type

`object`

.status.variables[].definitions[].schema.openAPIV3Schema.x-metadata.labels

Description

labels is a map of string keys and values that can be used to organize and categorize (scope and select) variables.

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterclasses`
  - `DELETE`: delete collection of ClusterClass
  - `GET`: list objects of kind ClusterClass
  - `POST`: create a new ClusterClass
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterclasses/{name}`
  - `DELETE`: delete the specified ClusterClass
  - `GET`: read the specified ClusterClass
  - `PATCH`: partially update the specified ClusterClass
  - `PUT`: replace the specified ClusterClass
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterclasses/{name}/status`
  - `GET`: read status of the specified ClusterClass
  - `PATCH`: partially update status of the specified ClusterClass
  - `PUT`: replace status of the specified ClusterClass

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterclasses

HTTP method

`DELETE`

Description

delete collection of ClusterClass

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind ClusterClass

HTTP responses

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

HTTP method

`POST`

Description

create a new ClusterClass

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

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
`body`	`ClusterClass` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterclasses/{name}

HTTP method

`DELETE`

Description

delete the specified ClusterClass

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 ClusterClass

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ClusterClass

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified ClusterClass

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`ClusterClass` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterclasses/{name}/status

HTTP method

`GET`

Description

read status of the specified ClusterClass

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ClusterClass

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified ClusterClass

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`ClusterClass` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ClusterClass` schema

HTTP code	Response body
201 - Created	`ClusterClass` schema
401 - Unauthorized	Empty

Machine [cluster.x-k8s.io/v1beta1]

Description

Machine is the Schema for the machines API.

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of Machine.
`status`	`object`	status is the observed state of Machine.

.spec

Description

spec is the desired state of Machine.

Type

`object`

Required

`bootstrap` `clusterName` `infrastructureRef`

Property	Type	Description
`bootstrap`	`object`	bootstrap is a reference to a local struct which encapsulates fields to configure the Machine’s bootstrapping mechanism.
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.
`failureDomain`	`string`	failureDomain is the failure domain the machine will be created in. Must match a key in the FailureDomains map stored on the cluster object.
`infrastructureRef`	`object`	infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time

Property	Type	Description
		limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.

Property	Type	Description
`providerID`	`string`	providerID is the identification ID of the machine provided by the provider. This field must match the provider ID as seen on the node object corresponding to this machine. This field is required by higher level consumers of cluster-api. Example use case is cluster autoscaler with cluster-api as provider. Clean-up logic in the autoscaler compares machines to nodes to find out machines at provider which could not get registered as Kubernetes nodes. With cluster-api as a generic out-of-tree provider for autoscaler, this field is required by autoscaler to be able to have a provider view of the list of machines. Another list of nodes is queried from the k8s apiserver and then a comparison is done to find out unregistered machines and are marked for delete. This field will be set by the actuators and consumed by higher level entities like autoscaler that will be interfacing with cluster-api as generic provider.
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. by Cluster API control plane providers to extend the semantic of the Ready condition for the Machine they control, like the kubeadm control provider adding ReadinessGates for

Property	Type	Description
		the APIServerPodHealthy, SchedulerPodHealthy conditions, etc. Another example are external controllers, e.g. responsible to install special software/hardware on the Machines; they can include the status of those components with a new condition and add this condition to ReadinessGates. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: In case readinessGates conditions start with the APIServer, ControllerManager, Scheduler prefix, and all those readiness gates condition are reporting the same message, when computing the Machine's Ready condition those readinessGates will be replaced by a single entry reporting "Control plane components: " + message. This helps to improve readability of conditions bubbling up to the Machine's owner resource / to the Cluster).

Property	Type	Description
`taints`	`array`	The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints
`version`	`string`	version defines the desired Kubernetes version. This field is meant to be optionally used by bootstrap providers.

.spec.bootstrap

Description

bootstrap is a reference to a local struct which encapsulates fields to configure the Machine’s bootstrapping mechanism.

Type

`object`

Property	Type	Description
`configRef`	`object`	configRef is a reference to a bootstrap provider-specific resource that holds configuration details. The reference is optional to allow users/operators to specify Bootstrap.DataSecretName without the need of a controller.
`dataSecretName`	`string`	dataSecretName is the name of the secret that stores the bootstrap data script. If nil, the Machine should remain in the Pending state.

.spec.bootstrap.configRef

Description

configRef is a reference to a bootstrap provider-specific resource that holds configuration details. The reference is optional to allow users/operators to specify Bootstrap.DataSecretName without the need of a controller.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this

Property	Type	Description
		would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗

Property	Type	Description
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.infrastructureRef

Description

infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

Property	Type	Description
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. by Cluster API control plane providers to extend the

semantic of the Ready condition for the Machine they control, like the kubeadm control provider adding ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc. Another example are external controllers, e.g. responsible to install special software/hardware on the Machines; they can include the status of those components with a new condition and add this condition to ReadinessGates. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: In case readinessGates conditions start with the APIServer, ControllerManager, Scheduler prefix, and all those readiness gates condition are reporting the same message, when computing the Machine's Ready condition those readinessGates will be replaced by a single entry reporting "Control plane components: " + message. This helps to improve readability of conditions bubbling up to the Machine's owner resource / to the Cluster).

Type

`array`

.spec.readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.

Property	Type	Description
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.taints

Description

The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints

Type

`array`

.spec.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.status

Description

status is the observed state of Machine.

Type

`object`

Property	Type	Description
`addresses`	`array`	addresses is a list of addresses assigned to the machine. This field is copied from the infrastructure provider reference.
`bootstrapReady`	`boolean`	bootstrapReady is the state of the bootstrap provid

Property	Type	Description
`certificatesExpiryDate`	`string`	certificatesExpiryDate is the expiry date of the machine certificates. This value is only set for contr plane machines.
`conditions`	`array`	conditions defines current service state of the Machine.
`deletion`	`object`	deletion contains information relating to removal of the Machine. Only present when the Machine has a deletionTimestamp and drain or wait for volume detach started.
`failureMessage`	`string`	failureMessage will be set in the event that there is terminal problem reconciling the Machine and will contain a more verbose string suitable for logging and human consumption. This field should not be set for transitive errors that controller faces that are expected to be fixed automatically over time (like service outages), but instead indicate that something is fundamentally wrong with the Machine's spec or the configuration the controller, and that manual intervention is required. Examples of terminal errors would be invalid combinations of settings in the spec, values that are unsupported by the controller, or the responsible controller itself being critically misconfigured. Any transient errors that occur during the reconciliation of Machines can be added as events

Property	Type	Description
		the Machine object and/or logged in the controller's output. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`failureReason`	`string`	failureReason will be set in the event that there is a terminal problem reconciling the Machine and will contain a succinct value suitable for machine interpretation. This field should not be set for transitive errors that controller faces that are expected to be fixed automatically over time (like service outages), but instead indicate that something is fundamentally wrong with the Machine's spec or the configuration the controller, and that manual intervention is required. Examples of terminal errors would be invalid combinations of settings in the spec, values that are unsupported by the controller, or the responsible controller itself being critically misconfigured. Any transient errors that occur during the reconciliation of Machines can be added as events the Machine object and/or logged in the controller's output. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-

Property	Type	Description
		api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`infrastructureReady`	`boolean`	infrastructureReady is the state of the infrastructure provider.
`lastUpdated`	`string`	lastUpdated identifies when the phase of the Machine last transitioned.
`nodeInfo`	`object`	nodeInfo is a set of ids/uuids to uniquely identify th node. More info: https://kubernetes.io/docs/concepts/nodes/node/#i ↗
`nodeRef`	`object`	nodeRef will point to the corresponding Node if it exists.
`observedGeneration`	`integer`	observedGeneration is the latest generation observed by the controller.
`phase`	`string`	phase represents the current phase of machine actuation.

Property	Type	Description
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in Machine's status with the V1Beta2 version.

.status.addresses

Description

addresses is a list of addresses assigned to the machine. This field is copied from the infrastructure provider reference.

Type

`array`

.status.addresses[]

Description

MachineAddress contains information for the node's address.

Type

`object`

Required

`address` `type`

Property	Type	Description
`address`	`string`	address is the machine address.
`type`	`string`	type is the machine address type, one of Hostname, ExternalIP, InternalIP, ExternalDNS or InternalDNS.

.status.conditions

Description

conditions defines current service state of the Machine.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

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

Property	Type	Description
		choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.deletion

Description

deletion contains information relating to removal of the Machine. Only present when the Machine has a deletionTimestamp and drain or wait for volume detach started.

Type

`object`

Property	Type	Description
`nodeDrainStartTime`	`string`	nodeDrainStartTime is the time when the drain of the node started and is used to determine if the NodeDrainTimeout is exceeded. Only present when the Machine has a deletionTimestamp and draining the node had been started.
`waitForNodeVolumeDetachStartTime`	`string`	waitForNodeVolumeDetachStartTime is the time when waiting for volume detachment started and is used to determine if the NodeVolumeDetachTimeout is exceeded. Detaching volumes from nodes is usually done by CSI implementations and the current state is observed from the node's `.Status.VolumesAttached` field. Only present when the Machine has a deletionTimestamp and waiting for volume detachments had been started.

.status.nodeInfo

Description

nodeInfo is a set of ids/uuids to uniquely identify the node. More info: https://kubernetes.io/docs/concepts/nodes/node/#info

Type

`object`

Required

`architecture` `bootID` `containerRuntimeVersion` `kernelVersion` `kubeProxyVersion` `kubeletVersion` `machineID` `operatingSystem` `osImage` `systemUUID`

Property	Type	Description
`architecture`	`string`	The Architecture reported by the node
`bootID`	`string`	Boot ID reported by the node.
`containerRuntimeVersion`	`string`	ContainerRuntime Version reported by the node through runtime remote API (e.g. containerd://1.4.
`kernelVersion`	`string`	Kernel Version reported by the node from 'uname (e.g. 3.16.0-0.bpo.4-amd64).
`kubeProxyVersion`	`string`	Deprecated: KubeProxy Version reported by the n
`kubeletVersion`	`string`	Kubelet Version reported by the node.
`machineID`	`string`	MachineID reported by the node. For unique macl identification in the cluster this field is preferred. L more from man(5) machine-id: http://man7.org/linux/man-pages/man5/machine-id.5.html↗

Property	Type	Description
`operatingSystem`	`string`	The Operating System reported by the node
`osImage`	`string`	OS Image reported by the node from /etc/os-relea (e.g. Debian GNU/Linux 7 (wheezy)).
`systemUUID`	`string`	SystemUUID reported by the node. For unique ma identification MachineID is preferred. This field is specific to Red Hat hosts https://access.redhat.com/documentation/en-us/red_hat_subscription_management/1/html/rhsr ↗

.status.nodeRef

Description

nodeRef will point to the corresponding Node if it exists.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if

Property	Type	Description
		the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-

Property	Type	Description
		with-objects/names/#uids ↗

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in Machine's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions represents the observations of a Machine's current state. Known condition types are Available, Ready, UpToDate, BootstrapConfigReady, InfrastructureReady, NodeReady, NodeHealthy, Deleting, Paused. If a MachineHealthCheck is targeting this machine, also HealthCheckSucceeded, OwnerRemediated conditions are added. Additionally control plane Machines controlled by KubeadmControlPlane will have following additional conditions: APIServerPodHealthy, ControllerManagerPodHealthy, SchedulerPodHealthy, EtcdPodHealthy, EtcdMemberHealthy.

.status.v1beta2.conditions

Description

conditions represents the observations of a Machine's current state. Known condition types are Available, Ready, UpToDate, BootstrapConfigReady, InfrastructureReady, NodeReady, NodeHealthy, Deleting, Paused. If a MachineHealthCheck is targeting this machine, also

HealthCheckSucceeded, OwnerRemediated conditions are added. Additionally control plane Machines controlled by KubeadmControlPlane will have following additional conditions: APIServerPodHealthy, ControllerManagerPodHealthy, SchedulerPodHealthy, EtcdPodHealthy, EtcdMemberHealthy.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation

Property	Type	Description
		is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machines`
  - `DELETE`: delete collection of Machine
  - `GET`: list objects of kind Machine
  - `POST`: create a new Machine

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machines/{name}`
  - `DELETE`: delete the specified Machine
  - `GET`: read the specified Machine
  - `PATCH`: partially update the specified Machine
  - `PUT`: replace the specified Machine
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machines/{name}/status`
  - `GET`: read status of the specified Machine
  - `PATCH`: partially update status of the specified Machine
  - `PUT`: replace status of the specified Machine

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machines

HTTP method

`DELETE`

Description

delete collection of Machine

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind Machine

HTTP responses

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

HTTP method

`POST`

Description

create a new Machine

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`Machine` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machines/{name}

HTTP method

`DELETE`

Description

delete the specified Machine

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 Machine

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified Machine

Query parameters

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified Machine

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Machine` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machines/{name}/status

HTTP method

`GET`

Description

read status of the specified Machine

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified Machine

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the

Parameter	Type	Description
		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	`Machine` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Machine

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Machine` schema	`application/json` formatted

HTTP responses

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

MachineSet [cluster.x-k8s.io/v1beta1]

Description

MachineSet is the Schema for the machinesets API.

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of MachineSet.
`status`	`object`	status is the observed state of MachineSet.

.spec

Description

spec is the desired state of MachineSet.

Type

`object`

Required

`clusterName` `selector`

Property	Type	Description
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.
`deletePolicy`	`string`	deletePolicy defines the policy used to identify nodes to delete when downscaling. Defaults to "Random". Valid values are "Random", "Newest", "Oldest"
`machineNamingStrategy`	`object`	machineNamingStrategy allows changing the naming pattern used when creating Machines. Note: InfraMachines & BootstrapConfigs will use the same name as the corresponding Machines.
`minReadySeconds`	`integer`	minReadySeconds is the minimum number of seconds for which a Node for a newly created machine should be ready before considering the replica available. Defaults to 0 (machine will be considered available as soon as the Node is ready)
`replicas`	`integer`	replicas is the number of desired replicas. This is a pointer to distinguish between explicit zero and unspecified. Defaults to: • if the Kubernetes autoscaler min size and max size annotations are set: • if it's a new MachineSet, use min size

Property	Type	Description
		• if the replicas field of the old MachineSet is < min size, use min size • if the replicas field of the old MachineSet is > max size, use max size • if the replicas field of the old MachineSet is in the (min size, max size) range, keep the value from the oldMS • otherwise use 1 Note: Defaulting will be run whenever the replicas field is not set: • A new MachineSet is created with replicas not se • On an existing MachineSet the replicas field was first set and is now unset. Those cases are especially relevant for the following Kubernetes autoscaler use cases: • A new MachineSet is created and replicas should be managed by the autoscaler • An existing MachineSet which initially wasn't controlled by the autoscaler should be later controlled by the autoscaler
`selector`	`object`	selector is a label query over machines that should match the replica count. Label keys and values that must match in order to be controlled by this MachineSet. It must match the machine template's labels. More info: https://kubernetes.io/docs/concepts/overview/workin with-objects/labels/#label-selectors ↗
`template`	`object`	template is the object that describes the machine tha will be created if insufficient replicas are detected.

Property	Type	Description
		Object references to custom resources are treated a templates.

.spec.machineNamingStrategy

Description

machineNamingStrategy allows changing the naming pattern used when creating Machines. Note: InfraMachines & BootstrapConfigs will use the same name as the corresponding Machines.

Type

`object`

Property	Type	Description
`template`	`string`	template defines the template to use for generating the names of the Machine objects. If not defined, it will fallback to `{{ .machineSet.name }}-{{ .random }}`. If the generated name string exceeds 63 characters, it will be trimmed to 58 characters and will get concatenated with a random suffix of length 5. Length of the template string must not exceed 256 characters. The template allows the following variables `.cluster.name`, `.machineSet.name` and `.random`. The variable `.cluster.name` retrieves the name of the cluster object that owns the Machines being created. The variable `.machineSet.name` retrieves the name of the MachineSet object that owns the Machines being created. The variable `.random` is substituted with random alphanumeric string, without vowels, of length 5. This variable is required part of the template. If not provided, validation will fail.

.spec.selector

Description

selector is a label query over machines that should match the replica count. Label keys and values that must match in order to be controlled by this MachineSet. It must match the machine template's labels. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors

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.selector.matchExpressions

Description

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

Type

`array`

.spec.selector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.selector.matchExpressions[].values

Description

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

Type

`array`

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

Type

`string`

.spec.selector.matchLabels

Description

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

Type

`object`

.spec.template

Description

template is the object that describes the machine that will be created if insufficient replicas are detected. Object references to custom resources are treated as templates.

Type

`object`

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

.spec.template.spec

Description

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

Type

`object`

Required

`bootstrap` `clusterName` `infrastructureRef`

Property	Type	Description
`bootstrap`	`object`	bootstrap is a reference to a local struct which encapsulates fields to configure the Machine’s bootstrapping mechanism.
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.
`failureDomain`	`string`	failureDomain is the failure domain the machine will be created in. Must match a key in the FailureDomains map stored on the cluster object.
`infrastructureRef`	`object`	infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.

Property	Type	Description
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.
`providerID`	`string`	providerID is the identification ID of the machine provided by the provider. This field must match the provider ID as seen on the node object corresponding to this machine. This field is required by higher level consumers of cluster-api. Example use case is cluster autoscaler with cluster-api as provider. Clean-up logic in the autoscaler compares machines to nodes to find out machines at provider which could not get

Property	Type	Description
		registered as Kubernetes nodes. With cluster-api as a generic out-of-tree provider for autoscaler, this field is required by autoscaler to be able to have a provider view of the list of machines. Another list of nodes is queried from the k8s apiserver and then a comparison is done to find out unregistered machines and are marked for delete. This field will be set by the actuators and consumed by higher level entities like autoscaler that will be interfacing with cluster-api as generic provider.
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. by Cluster API control plane providers to extend the semantic of the Ready condition for the Machine they control, like the kubeadm control provider adding ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc. Another example are external controllers, e.g. responsible to install special software/hardware on the Machines; they can include the status of those components with a new condition and add this condition to ReadinessGates. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: In case readinessGates conditions start with

Property	Type	Description
		the APIServer, ControllerManager, Scheduler prefix, and all those readiness gates condition are reporting the same message, when computing the Machine's Ready condition those readinessGates will be replaced by a single entry reporting "Control plane components: " + message. This helps to improve readability of conditions bubbling up to the Machine's owner resource / to the Cluster).
`taints`	`array`	The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints
`version`	`string`	version defines the desired Kubernetes version. This field is meant to be optionally used by bootstrap providers.

.spec.template.spec.bootstrap

Description

bootstrap is a reference to a local struct which encapsulates fields to configure the Machine’s bootstrapping mechanism.

Type

`object`

Property	Type	Description
`configRef`	`object`	configRef is a reference to a bootstrap provider-specific resource that holds configuration details. The reference is optional to allow users/operators to specify Bootstrap.DataSecretName without the need of a controller.
`dataSecretName`	`string`	dataSecretName is the name of the secret that stores the bootstrap data script. If nil, the Machine should remain in the Pending state.

.spec.template.spec.bootstrap.configRef

Description

configRef is a reference to a bootstrap provider-specific resource that holds configuration details. The reference is optional to allow users/operators to specify Bootstrap.DataSecretName without the need of a controller.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field

Property	Type	Description
		access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗

Property	Type	Description
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.template.spec.infrastructureRef

Description

infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

Property	Type	Description
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.template.spec.readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. by Cluster API control plane providers to extend the

semantic of the Ready condition for the Machine they control, like the kubeadm control provider adding ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc. Another example are external controllers, e.g. responsible to install special software/hardware on the Machines; they can include the status of those components with a new condition and add this condition to ReadinessGates. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: In case readinessGates conditions start with the APIServer, ControllerManager, Scheduler prefix, and all those readiness gates condition are reporting the same message, when computing the Machine's Ready condition those readinessGates will be replaced by a single entry reporting "Control plane components: " + message. This helps to improve readability of conditions bubbling up to the Machine's owner resource / to the Cluster).

Type

`array`

.spec.template.spec.readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.

Property	Type	Description
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.template.spec.taints

Description

The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints

Type

`array`

.spec.template.spec.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.status

Description

status is the observed state of MachineSet.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the number of available replicas (ready for at least minReadySeconds) for this MachineSet.

Property	Type	Description
`conditions`	`array`	conditions defines current service state of the MachineSet.
`failureMessage`	`string`	failureMessage will be set in the event that there is a terminal problem reconciling the Machine and will contain a more verbose string suitable for logging and human consumption. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`failureReason`	`string`	failureReason will be set in the event that there is a terminal problem reconciling the Machine and will contain a succinct value suitable for machine interpretation. In the event that there is a terminal problem reconciling the replicas, both FailureReason and FailureMessage will be set. FailureReason will be populated with a succinct value suitable for machine interpretation, while FailureMessage will contain a more verbose string suitable for logging and human consumption. These fields should not be set for transitive errors that a controller faces that are expected to be fixed automatically over time (like service

Property	Type	Description
		outages), but instead indicate that something is fundamentally wrong with the MachineTemplate's spec or the configuration of the machine controller, and that manual intervention is required. Examples of terminal errors would be invalid combinations of settings in the spec, values that are unsupported by the machine controller, or the responsible machine controller itself being critically misconfigured. Any transient errors that occur during the reconciliation of Machines can be added as events to the MachineSet object and/or logged in the controller's output. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.

Property	Type	Description
`fullyLabeledReplicas`	`integer`	fullyLabeledReplicas is the number of replicas that have labels matching the labels of the machine template of the MachineSet. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`observedGeneration`	`integer`	observedGeneration reflects the generation of the most recently observed MachineSet.
`readyReplicas`	`integer`	readyReplicas is the number of ready replicas for this MachineSet. A machine is considered ready when the node has been created and is "Ready".
`replicas`	`integer`	replicas is the most recently observed number of replicas.

Property	Type	Description
`selector`	`string`	selector is the same as the label selector but in the string format to avoid introspection by clients. The string will be in the same format as the query-param syntax. More info about label selectors: http://kubernetes.io/docs/user-guide/labels#label-selectors ↗
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in MachineSet's status with the V1Beta2 version.

.status.conditions

Description

conditions defines current service state of the MachineSet.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across

Property	Type	Description
		resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in MachineSet's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the number of available replicas for this MachineSet. A machine is considered available when Machine's Available condition is true.
`conditions`	`array`	conditions represents the observations of a MachineSet's current state. Known condition types are MachinesReady, MachinesUpToDate, ScalingUp, ScalingDown, Remediating, Deleting, Paused.

Property	Type	Description
`readyReplicas`	`integer`	readyReplicas is the number of ready replicas for this MachineSet. A machine is considered ready when Machine's Ready condition is true.
`upToDateReplicas`	`integer`	upToDateReplicas is the number of up-to-date replicas for this MachineSet. A machine is considered up-to-date when Machine's UpToDate condition is true.

.status.v1beta2.conditions

Description

conditions represents the observations of a MachineSet's current state. Known condition types are MachinesReady, MachinesUpToDate, ScalingUp, ScalingDown, Remediating, Deleting, Paused.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinesets`
  - `DELETE`: delete collection of MachineSet
  - `GET`: list objects of kind MachineSet
  - `POST`: create a new MachineSet
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinesets/{name}`
  - `DELETE`: delete the specified MachineSet
  - `GET`: read the specified MachineSet
  - `PATCH`: partially update the specified MachineSet
  - `PUT`: replace the specified MachineSet
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinesets/{name}/status`
  - `GET`: read status of the specified MachineSet
  - `PATCH`: partially update status of the specified MachineSet
  - `PUT`: replace status of the specified MachineSet

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinesets/{name}/scale`
  - `GET`: read scale of the specified MachineSet
  - `PATCH`: partially update scale of the specified MachineSet
  - `PUT`: replace scale of the specified MachineSet

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinesets

HTTP method

`DELETE`

Description

delete collection of MachineSet

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind MachineSet

HTTP responses

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

HTTP method

`POST`

Description

create a new MachineSet

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`MachineSet` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinesets/{name}

HTTP method

`DELETE`

Description

delete the specified MachineSet

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

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified MachineSet

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified MachineSet

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

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	`MachineSet` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified MachineSet

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently

Parameter	Type	Description
		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`	`MachineSet` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinesets/{name}/status

HTTP method

`GET`

Description

read status of the specified MachineSet

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified MachineSet

Query parameters

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

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified MachineSet

Query parameters

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

Parameter	Type	Description
		request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`MachineSet` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinesets/{name}/scale

HTTP method

`GET`

Description

read scale of the specified MachineSet

HTTP responses

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

HTTP method

`PATCH`

Description

partially update scale of the specified MachineSet

Query parameters

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

HTTP method

`PUT`

Description

replace scale of the specified MachineSet

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`Scale` schema	`application/json` formatted

MachineDeployment [cluster.x-k8s.io/v1beta1]

Description

MachineDeployment is the Schema for the machinedeployments 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:

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

.spec

Description

spec is the desired state of MachineDeployment.

Type

`object`

Required

`clusterName` `selector` `template`

Property	Type	Description
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.
`machineNamingStrategy`	`object`	machineNamingStrategy allows changing the naming pattern used when creating

Property	Type	Description
		Machines. Note: InfraMachines & BootstrapConfigs will use the same name as the corresponding Machines.
`minReadySeconds`	`integer`	minReadySeconds is the minimum number of seconds for which a Node for a newly created machine should be ready before considering the replica available. Defaults to 0 (machine will be considered available as soon as the Node is ready)
`paused`	`boolean`	paused indicates that the deployment is paused.
`progressDeadlineSeconds`	`integer`	progressDeadlineSeconds is the maximum time in seconds for a deployment to make progress before it is considered to be failed. The deployment controller will continue to process failed deployments and a condition with a ProgressDeadlineExceeded reason will be surfaced in the deployment status. Note that progress will not be estimated during the time a deployment is paused. Defaults to 600s. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/issues/11470↗ for more details.

Property	Type	Description

`replicas`	`integer`	replicas is the number of desired machines. This is a pointer to distinguish between explicit zero and not specified.
		
		Defaults to:
		
		• if the Kubernetes autoscaler min size and max size annotations are set:
		  • if it's a new MachineDeployment, use min size
		  • if the replicas field of the old MachineDeployment is < min size, use min size
		  • if the replicas field of the old MachineDeployment is > max size, use max size
		  • if the replicas field of the old MachineDeployment is in the (min size, max size) range, keep the value from the oldMD
		• otherwise use 1 Note: Defaulting will be run whenever the replicas field is not set:
		• A new MachineDeployment is created with replicas not set.
		• On an existing MachineDeployment the replicas field was first set and is now unset. Those cases are especially relevant for the following Kubernetes autoscaler use cases:
		• A new MachineDeployment is created and replicas should be managed by the autoscaler

Property	Type	Description
		• An existing MachineDeployment which initially wasn't controlled by the autoscaler should be later controlled by the autoscaler
`revisionHistoryLimit`	`integer`	revisionHistoryLimit is the number of old MachineSets to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/issues/10479 for more details.
`rolloutAfter`	`string`	rolloutAfter is a field to indicate a rollout should be performed after the specified time even if no changes have been made to the MachineDeployment. Example: In the YAML the time can be specified in the RFC3339 format. To specify the rolloutAfter target as March 9, 2023, at 9 am UTC use "2023-03-09T09:00:00Z".
`selector`	`object`	selector is the label selector for machines. Existing MachineSets whose machines are selected by this will be the ones affected by this deployment. It must match the machine template's labels.

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

.spec.machineNamingStrategy

Description

machineNamingStrategy allows changing the naming pattern used when creating Machines. Note: InfraMachines & BootstrapConfigs will use the same name as the corresponding Machines.

Type

`object`

Property	Type	Description
`template`	`string`	template defines the template to use for generating the names of the Machine objects. If not defined, it will fallback to `{{ .machineSet.name }}-{{ .random }}`. If the generated name string exceeds 63 characters, it will be trimmed to 58 characters and will get concatenated with a random suffix of length 5. Length of the template string must not exceed 256 characters. The template allows the following variables `.cluster.name`, `.machineSet.name` and `.random`. The variable `.cluster.name` retrieves the name of the cluster object that owns the Machines being created. The variable `.machineSet.name` retrieves the name of the

Property	Type	Description
		MachineSet object that owns the Machines being created. The variable `.random` is substituted with random alphanumeric string, without vowels, of length 5. This variable is required part of the template. If not provided, validation will fail.

.spec.selector

Description

selector is the label selector for machines. Existing MachineSets whose machines are selected by this will be the ones affected by this deployment. It must match the machine template's labels.

Type

`object`

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

.spec.selector.matchExpressions

Description

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

Type

`array`

.spec.selector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.selector.matchExpressions[].values

Description

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

Type

`array`

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

Type

`string`

.spec.selector.matchLabels

Description

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

Type

`object`

.spec.strategy

Description

strategy is the deployment strategy to use to replace existing machines with new ones.

Type

`object`

Property	Type	Description
`remediation`	`object`	remediation controls the strategy of remediating unhealthy machines and how remediating operations should occur during the lifecycle of the dependant MachineSets.
`rollingUpdate`	`object`	rollingUpdate is the rolling update config params. Present only if MachineDeploymentStrategyType = RollingUpdate.
`type`	`string`	type of deployment. Allowed values are RollingUpdate and OnDelete. The default is RollingUpdate.

.spec.strategy.remediation

Description

remediation controls the strategy of remediating unhealthy machines and how remediating operations should occur during the lifecycle of the dependant MachineSets.

Type

`object`

Property	Type	Description
`maxInFlight`	`integer|string`	maxInFlight determines how many in flight remediations should happen at the same time. Remediation only happens on the MachineSet with the most current revision, while older MachineSets (usually present during rollout operations) aren't allowed to remediate.

Property	Type	Description
		Note: In general (independent of remediations), unhealthy machines are always prioritized during scale down operations over healthy ones. MaxInFlight can be set to a fixed number or a percentage. Example: when this is set to 20%, the MachineSet controller deletes at most 20% of the desired replicas. If not set, remediation is limited to all machines (bounded by replicas) under the active MachineSet's management.

.spec.strategy.rollingUpdate

Description

rollingUpdate is the rolling update config params. Present only if MachineDeploymentStrategyType = RollingUpdate.

Type

`object`

Property	Type	Description
`deletePolicy`	`string`	deletePolicy defines the policy used by the MachineDeployment to identify nodes to delete when downscaling. Valid values are "Random", "Newest", "Oldest" When no value is supplied, the default DeletePolicy of MachineSet is used
`maxSurge`	`integer|string`	maxSurge is the maximum number of machines that can be scheduled above the desired number of

Property	Type	Description
		machines. Value can be an absolute number (ex: 5) or a percentage of desired machines (ex: 10%). This can not be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up. Defaults to 1. Example: when this is set to 30%, the new MachineSet can be scaled up immediately when the rolling update starts, such that the total number of old and new machines do not exceed 130% of desired machines. Once old machines have been killed, new MachineSet can be scaled up further, ensuring that total number of machines running at any time during the update is at most 130% of desired machines.
`maxUnavailable`	`integer|string`	maxUnavailable is the maximum number of machines that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of desired machines (ex: 10%). Absolute number is calculated from percentage by rounding down. This can not be 0 if MaxSurge is 0. Defaults to 0. Example: when this is set to 30%, the old MachineSet can be scaled down to 70% of desired machines immediately when the rolling update starts. Once new machines are ready, old MachineSet can be scaled down further, followed by scaling up the new MachineSet, ensuring that the total number of machines available at all times during the update is at least 70% of desired machines.

.spec.template

Description

template describes the machines that will be created.

Type

`object`

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

.spec.template.spec

Description

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

Type

`object`

Required

`bootstrap` `clusterName` `infrastructureRef`

Property	Type	Description
`bootstrap`	`object`	bootstrap is a reference to a local struct which encapsulates fields to configure the Machine’s bootstrapping mechanism.
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.
`failureDomain`	`string`	failureDomain is the failure domain the machine will be created in. Must match a key in the FailureDomains map stored on the cluster object.
`infrastructureRef`	`object`	infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time

Property	Type	Description
		limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.

Property	Type	Description
`providerID`	`string`	providerID is the identification ID of the machine provided by the provider. This field must match the provider ID as seen on the node object corresponding to this machine. This field is required by higher level consumers of cluster-api. Example use case is cluster autoscaler with cluster-api as provider. Clean-up logic in the autoscaler compares machines to nodes to find out machines at provider which could not get registered as Kubernetes nodes. With cluster-api as a generic out-of-tree provider for autoscaler, this field is required by autoscaler to be able to have a provider view of the list of machines. Another list of nodes is queried from the k8s apiserver and then a comparison is done to find out unregistered machines and are marked for delete. This field will be set by the actuators and consumed by higher level entities like autoscaler that will be interfacing with cluster-api as generic provider.
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. by Cluster API control plane providers to extend the semantic of the Ready condition for the Machine they control, like the kubeadm control provider adding ReadinessGates for

Property	Type	Description
		the APIServerPodHealthy, SchedulerPodHealthy conditions, etc. Another example are external controllers, e.g. responsible to install special software/hardware on the Machines; they can include the status of those components with a new condition and add this condition to ReadinessGates. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: In case readinessGates conditions start with the APIServer, ControllerManager, Scheduler prefix, and all those readiness gates condition are reporting the same message, when computing the Machine's Ready condition those readinessGates will be replaced by a single entry reporting "Control plane components: " + message. This helps to improve readability of conditions bubbling up to the Machine's owner resource / to the Cluster).

Property	Type	Description
`taints`	`array`	The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints
`version`	`string`	version defines the desired Kubernetes version. This field is meant to be optionally used by bootstrap providers.

.spec.template.spec.bootstrap

Description

bootstrap is a reference to a local struct which encapsulates fields to configure the Machine’s bootstrapping mechanism.

Type

`object`

Property	Type	Description
`configRef`	`object`	configRef is a reference to a bootstrap provider-specific resource that holds configuration details. The reference is optional to allow users/operators to specify Bootstrap.DataSecretName without the need of a controller.
`dataSecretName`	`string`	dataSecretName is the name of the secret that stores the bootstrap data script. If nil, the Machine should remain in the Pending state.

.spec.template.spec.bootstrap.configRef

Description

configRef is a reference to a bootstrap provider-specific resource that holds configuration details. The reference is optional to allow users/operators to specify Bootstrap.DataSecretName without the need of a controller.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this

Property	Type	Description
		would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗

Property	Type	Description
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.template.spec.infrastructureRef

Description

infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

Property	Type	Description
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.template.spec.readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. by Cluster API control plane providers to extend the

semantic of the Ready condition for the Machine they control, like the kubeadm control provider adding ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc. Another example are external controllers, e.g. responsible to install special software/hardware on the Machines; they can include the status of those components with a new condition and add this condition to ReadinessGates. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: In case readinessGates conditions start with the APIServer, ControllerManager, Scheduler prefix, and all those readiness gates condition are reporting the same message, when computing the Machine's Ready condition those readinessGates will be replaced by a single entry reporting "Control plane components: " + message. This helps to improve readability of conditions bubbling up to the Machine's owner resource / to the Cluster).

Type

`array`

.spec.template.spec.readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.

Property	Type	Description
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.template.spec.taints

Description

The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints

Type

`array`

.spec.template.spec.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.status

Description

status is the observed state of MachineDeployment.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the total number of available machines (ready for at least minReadySeconds) targeted by this deployment.

Property	Type	Description
`conditions`	`array`	conditions defines current service state of the MachineDeployment.
`observedGeneration`	`integer`	observedGeneration is the generation observed by the deployment controller.
`phase`	`string`	phase represents the current phase of a MachineDeployment (ScalingUp, ScalingDown, Running, Failed, or Unknown).
`readyReplicas`	`integer`	readyReplicas is the total number of ready machines targeted by this deployment.
`replicas`	`integer`	replicas is the total number of non-terminated machines targeted by this deployment (their labels match the selector).
`selector`	`string`	selector is the same as the label selector but in the string format to avoid introspection by clients. The string will be in the same format as the query-param syntax. More info about label selectors: http://kubernetes.io/docs/user-guide/labels#label-selectors ↗
`unavailableReplicas`	`integer`	unavailableReplicas is the total number of unavailable machines targeted by this

Property	Type	Description
		deployment. This is the total number of machines that are still required for the deployment to have 100% available capacity. They may either be machines that are running but not yet available or machines that still have not been created. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`updatedReplicas`	`integer`	updatedReplicas is the total number of non-terminated machines targeted by this deployment that have the desired template spec.
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in MachineDeployment's status with the V1Beta2 version.

.status.conditions

Description

conditions defines current service state of the MachineDeployment.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in MachineDeployment's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the number of available replicas for this MachineDeployment. A machine is considered available when Machine's Available condition is true.
`conditions`	`array`	conditions represents the observations of a MachineDeployment's current state. Known

Property	Type	Description
		condition types are Available, MachinesReady, MachinesUpToDate, ScalingUp, ScalingDown, Remediating, Deleting, Paused.
`readyReplicas`	`integer`	readyReplicas is the number of ready replicas for this MachineDeployment. A machine is considered ready when Machine's Ready condition is true.
`upToDateReplicas`	`integer`	upToDateReplicas is the number of up-to-date replicas targeted by this deployment. A machine is considered up-to-date when Machine's UpToDate condition is true.

.status.v1beta2.conditions

Description

conditions represents the observations of a MachineDeployment's current state. Known condition types are Available, MachinesReady, MachinesUpToDate, ScalingUp, ScalingDown, Remediating, Deleting, Paused.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedeployments`
  - `DELETE`: delete collection of MachineDeployment
  - `GET`: list objects of kind MachineDeployment
  - `POST`: create a new MachineDeployment
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedeployments/{name}`
  - `DELETE`: delete the specified MachineDeployment
  - `GET`: read the specified MachineDeployment
  - `PATCH`: partially update the specified MachineDeployment
  - `PUT`: replace the specified MachineDeployment
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedeployments/{name}/status`
  - `GET`: read status of the specified MachineDeployment
  - `PATCH`: partially update status of the specified MachineDeployment
  - `PUT`: replace status of the specified MachineDeployment

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedeployments/{name}/scale`
  - `GET`: read scale of the specified MachineDeployment
  - `PATCH`: partially update scale of the specified MachineDeployment
  - `PUT`: replace scale of the specified MachineDeployment

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedeployments

HTTP method

`DELETE`

Description

delete collection of MachineDeployment

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind MachineDeployment

HTTP responses

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

HTTP method

`POST`

Description

create a new MachineDeployment

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`MachineDeployment` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedeployments/{name}

HTTP method

`DELETE`

Description

delete the specified MachineDeployment

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

HTTP code	Response body
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified MachineDeployment

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified MachineDeployment

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

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.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified MachineDeployment

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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`	`MachineDeployment` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedeployments/{name}/status

HTTP method

`GET`

Description

read status of the specified MachineDeployment

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified MachineDeployment

Query parameters

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

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified MachineDeployment

Query parameters

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

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`MachineDeployment` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedeployments/{name}/scale

HTTP method

`GET`

Description

read scale of the specified MachineDeployment

HTTP responses

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

HTTP method

`PATCH`

Description

partially update scale of the specified MachineDeployment

Query parameters

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

HTTP method

`PUT`

Description

replace scale of the specified MachineDeployment

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`Scale` schema	`application/json` formatted

MachinePool [cluster.x-k8s.io/v1beta1]

Description

MachinePool is the Schema for the machinepools API.

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of MachinePool.
`status`	`object`	status is the observed state of MachinePool.

.spec

Description

spec is the desired state of MachinePool.

Type

`object`

Required

`clusterName` `template`

Property	Type	Description
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.
`failureDomains`	`array`	failureDomains is the list of failure domains this MachinePool should be attached to.
`minReadySeconds`	`integer`	minReadySeconds is the minimum number of seconds for which a newly created machine instances should be ready. Defaults to 0 (machine instance will be considered available as soon as it is ready)
`providerIDList`	`array`	providerIDList are the identification IDs of machine instances provided by the provider. This field must match the provider IDs as seen on the node objects corresponding to a machine pool's machine instances.
`replicas`	`integer`	replicas is the number of desired machines. Defaults to 1. This is a pointer to distinguish between explicit zero and not specified.
`template`	`object`	template describes the machines that will be created.

.spec.failureDomains

Description

failureDomains is the list of failure domains this MachinePool should be attached to.

Type

`array`

.spec.failureDomains[]

Type

`string`

.spec.providerIDList

Description

providerIDList are the identification IDs of machine instances provided by the provider. This field must match the provider IDs as seen on the node objects corresponding to a machine pool's machine instances.

Type

`array`

.spec.providerIDList[]

Type

`string`

.spec.template

Description

template describes the machines that will be created.

Type

`object`

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

.spec.template.spec

Description

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

Type

`object`

Required

`bootstrap` `clusterName` `infrastructureRef`

Property	Type	Description
`bootstrap`	`object`	bootstrap is a reference to a local struct which encapsulates fields to configure the Machine’s bootstrapping mechanism.

Property	Type	Description
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.
`failureDomain`	`string`	failureDomain is the failure domain the machine will be created in. Must match a key in the FailureDomains map stored on the cluster object.
`infrastructureRef`	`object`	infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. Defaults to 10 seconds.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a node. The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend

Property	Type	Description
		on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.
`providerID`	`string`	providerID is the identification ID of the machine provided by the provider. This field must match the provider ID as seen on the node object corresponding to this machine. This field is required by higher level consumers of cluster-api. Example use case is cluster autoscaler with cluster-api as provider. Clean-up logic in the autoscaler compares machines to nodes to find out machines at provider which could not get registered as Kubernetes nodes. With cluster-api as a generic out-of-tree provider for autoscaler, this field is required by autoscaler to be able to have a provider view of the list of machines. Another list of nodes is queried from the k8s apiserver and then a comparison is done to find out unregistered machines and are marked for delete. This field will be set by the actuators and consumed by higher level entities like autoscaler that will be interfacing with cluster-api as generic provider.
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition.

Property	Type	Description
		This field can be used e.g. by Cluster API control plane providers to extend the semantic of the Ready condition for the Machine they control, like the kubeadm control provider adding ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc. Another example are external controllers, e.g. responsible to install special software/hardware on the Machines; they can include the status of those components with a new condition and add this condition to ReadinessGates. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: In case readinessGates conditions start with the APIServer, ControllerManager, Scheduler prefix, and all those readiness gates condition are reporting the same message, when computing the Machine's Ready condition those readinessGates will be replaced by a single entry reporting "Control plane components: " + message. This helps to improve readability of conditions bubbling up to the Machine's owner resource / to the Cluster).
`taints`	`array`	The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively

Property	Type	Description
		reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints
`version`	`string`	version defines the desired Kubernetes version. This field is meant to be optionally used by bootstrap providers.

.spec.template.spec.bootstrap

Description

bootstrap is a reference to a local struct which encapsulates fields to configure the Machine’s bootstrapping mechanism.

Type

`object`

Property	Type	Description
`configRef`	`object`	configRef is a reference to a bootstrap provider-specific resource that holds configuration details. The reference is optional to allow users/operators to specify Bootstrap.DataSecretName without the need of a controller.

Property	Type	Description
`dataSecretName`	`string`	dataSecretName is the name of the secret that stores the bootstrap data script. If nil, the Machine should remain in the Pending state.

.spec.template.spec.bootstrap.configRef

Description

configRef is a reference to a bootstrap provider-specific resource that holds configuration details. The reference is optional to allow users/operators to specify Bootstrap.DataSecretName without the need of a controller.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

Property	Type	Description
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.template.spec.infrastructureRef

Description

infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

Property	Type	Description
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.template.spec.readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition. This field can be used e.g. by Cluster API control plane providers to extend the semantic of the Ready condition for the Machine they control, like the kubeadm control provider adding ReadinessGates for the APIServerPodHealthy, SchedulerPodHealthy conditions, etc. Another example are external controllers, e.g. responsible to install special software/hardware on the Machines; they can include the status of those components with a new condition and add this condition to ReadinessGates. NOTE: This field is considered only for computing v1beta2 conditions. NOTE: In case readinessGates conditions start with the APIServer, ControllerManager, Scheduler prefix, and all those readiness gates condition are reporting the same message, when computing the Machine's Ready condition those readinessGates will be replaced by a single entry reporting "Control plane components: " + message. This helps to improve readability of conditions bubbling up to the Machine's owner resource / to the Cluster).

Type

`array`

.spec.template.spec.readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.template.spec.taints

Description

The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints

Type

`array`

.spec.template.spec.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.

Property	Type	Description
`value`	`string`	The taint value corresponding to the taint key.

.status

Description

status is the observed state of MachinePool.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the number of available replicas (ready for at least minReadySeconds) for this MachinePool.
`bootstrapReady`	`boolean`	bootstrapReady is the state of the bootstrap provider.
`conditions`	`array`	conditions define the current service state of the MachinePool.
`failureMessage`	`string`	failureMessage indicates that there is a problem reconciling the state, and will be set to a descriptive error message. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see

Property	Type	Description
		https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`failureReason`	`string`	failureReason indicates that there is a problem reconciling the state, and will be set to a token value suitable for programmatic interpretation. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`infrastructureReady`	`boolean`	infrastructureReady is the state of the infrastructure provider.
`nodeRefs`	`array`	nodeRefs will point to the corresponding Nodes if they exist.
`observedGeneration`	`integer`	observedGeneration is the latest generation observed by the controller.
`phase`	`string`	phase represents the current phase of cluster actuation.

Property	Type	Description
`readyReplicas`	`integer`	readyReplicas is the number of ready replicas for this MachinePool. A machine is considered ready when the node has been created and is "Ready".
`replicas`	`integer`	replicas is the most recently observed number of replicas.
`unavailableReplicas`	`integer`	unavailableReplicas is the total number of unavailable machine instances targeted by this machine pool. This is the total number of machine instances that are still required for the machine pool to have 100% available capacity. They may either be machine instances that are running but not yet available or machine instances that still have not been created. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md↗ for more details.
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in MachinePool's status with the V1Beta2 version.

.status.conditions

Description

conditions define the current service state of the MachinePool.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.

Property	Type	Description
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.nodeRefs

Description

nodeRefs will point to the corresponding Nodes if they exist.

Type

`array`

.status.nodeRefs[]

Description

ObjectReference contains enough information to let you inspect or modify the referred object.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

Property	Type	Description
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in MachinePool's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the number of available replicas for this MachinePool. A machine is considered available when Machine's Available condition is true.
`conditions`	`array`	conditions represents the observations of a MachinePool's current state. Known condition types are Available, BootstrapConfigReady, InfrastructureReady, MachinesReady, MachinesUpToDate, ScalingUp, ScalingDown, Remediating, Deleting, Paused.
`readyReplicas`	`integer`	readyReplicas is the number of ready replicas for this MachinePool. A machine is considered ready when Machine's Ready condition is true.
`upToDateReplicas`	`integer`	upToDateReplicas is the number of up-to-date replicas targeted by this MachinePool. A machine is considered up-to-date when Machine's UpToDate condition is true.

.status.v1beta2.conditions

Description

conditions represents the observations of a MachinePool's current state. Known condition types are Available, BootstrapConfigReady, InfrastructureReady, MachinesReady, MachinesUpToDate, ScalingUp, ScalingDown, Remediating, Deleting, Paused.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.

Property	Type	Description
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinepools`
  - `DELETE`: delete collection of MachinePool
  - `GET`: list objects of kind MachinePool
  - `POST`: create a new MachinePool
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinepools/{name}`
  - `DELETE`: delete the specified MachinePool
  - `GET`: read the specified MachinePool
  - `PATCH`: partially update the specified MachinePool

- `PUT`: replace the specified MachinePool

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinepools/{name}/status`
  - `GET`: read status of the specified MachinePool
  - `PATCH`: partially update status of the specified MachinePool
  - `PUT`: replace status of the specified MachinePool

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinepools/{name}/scale`
  - `GET`: read scale of the specified MachinePool
  - `PATCH`: partially update scale of the specified MachinePool
  - `PUT`: replace scale of the specified MachinePool

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinepools

HTTP method

`DELETE`

Description

delete collection of MachinePool

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind MachinePool

HTTP responses

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

HTTP method

`POST`

Description

create a new MachinePool

Query parameters

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`MachinePool` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinepools/{name}

HTTP method

`DELETE`

Description

delete the specified MachinePool

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 MachinePool

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified MachinePool

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified MachinePool

Query parameters

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`MachinePool` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinepools/{

name}/status

HTTP method

`GET`

Description

read status of the specified MachinePool

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified MachinePool

Query parameters

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

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified MachinePool

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the

Parameter	Type	Description
		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`	`MachinePool` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinepools/{name}/scale

HTTP method

`GET`

Description

read scale of the specified MachinePool

HTTP responses

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

HTTP method

`PATCH`

Description

partially update scale of the specified MachinePool

Query parameters

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

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

HTTP method

`PUT`

Description

replace scale of the specified MachinePool

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`Scale` schema	`application/json` formatted

MachineHealthCheck [cluster.x-k8s.io/v1beta1]

Description

MachineHealthCheck is the Schema for the machinehealthchecks 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:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the specification of machine health check policy
`status`	`object`	status is the most recently observed status of MachineHealthCheck resource

.spec

Description

spec is the specification of machine health check policy

Type

`object`

Required

`clusterName` `selector`

Property	Type	Description
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.

Property	Type	Description
`maxUnhealthy`	`integer|string`	maxUnhealthy specifies the maximum number of unhealthy machines allowed. Any further remediation is only allowed if at most "maxUnhealthy" machines selected by "selector" are not healthy. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/issues/10722↗ for more details.
`nodeStartupTimeout`	`string`	nodeStartupTimeout allows to set the maximum time for MachineHealthCheck to consider a Machine unhealthy if a corresponding Node isn't associated through a `Spec.ProviderID` field. The duration set in this field is compared to the greatest of: • Cluster's infrastructure ready condition timestamp (if and when available) • Control Plane's initialized condition timestamp (if and when available) • Machine's infrastructure ready condition timestamp (if and when available) • Machine's metadata creation timestamp Defaults to 10 minutes. If you wish to disable this feature, set the value explicitly to 0.

Property	Type	Description
`remediationTemplate`	`object`	remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.
`selector`	`object`	selector is a label selector to match machines whose health will be exercised
`unhealthyConditions`	`array`	unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.
`unhealthyRange`	`string`	unhealthyRange specifies the range of unhealthy machines allowed. Any further remediation is only allowed if the number of machines selected by "selector" as not healthy is within the range of "unhealthyRange". Takes precedence over maxUnhealthy. Eg. "[3-5]" - This means that remediation will be allowed only when: (a) there are at least 3 unhealthy machines (and) (b) there are at most 5 unhealthy machines

Property	Type	Description
		Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/issues/10722 for more details.

.spec.remediationTemplate

Description

remediationTemplate is a reference to a remediation template provided by an infrastructure provider. This field is completely optional, when filled, the MachineHealthCheck controller creates a new object from the template referenced and hands off remediation of the machine to a controller that lives outside of Cluster API.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.

Property	Type	Description
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-

Property	Type	Description
		architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.selector

Description

selector is a label selector to match machines whose health will be exercised

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.selector.matchExpressions

Description

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

Type

`array`

.spec.selector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.selector.matchExpressions[].values

Description

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

Type

`array`

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

Type

`string`

.spec.selector.matchLabels

Description

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

Type

`object`

.spec.unhealthyConditions

Description

unhealthyConditions contains a list of the conditions that determine whether a node is considered unhealthy. The conditions are combined in a logical OR, i.e. if any of the conditions is met, the node is unhealthy.

Type

`array`

.spec.unhealthyConditions[]

Description

UnhealthyCondition represents a Node condition type and value with a timeout specified as a duration. When the named condition has been in the given status for at least the timeout value, a node is considered unhealthy.

Type

`object`

Required

`status` `timeout` `type`

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`timeout`	`string`	timeout is the duration that a node must be in a given status for, after which the node is considered unhealthy. For example, with a value of "1h", the node must match the status for at least 1 hour before being considered unhealthy.
`type`	`string`	type of Node condition

.status

Description

status is the most recently observed status of MachineHealthCheck resource

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions defines current service state of the MachineHealthCheck.
`currentHealthy`	`integer`	currentHealthy is the total number of healthy machines counted by this machine health check
`expectedMachines`	`integer`	expectedMachines is the total number of machines counted by this machine health check
`observedGeneration`	`integer`	observedGeneration is the latest generation observed by the controller.
`remediationsAllowed`	`integer`	remediationsAllowed is the number of further remediations allowed by this machine health check before maxUnhealthy short circuiting will be applied
`targets`	`array`	targets shows the current list of machines the machine health check is watching
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in MachineHealthCheck's status with the V1Beta2 version.

.status.conditions

Description

conditions defines current service state of the MachineHealthCheck.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

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

Property	Type	Description
		choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.targets

Description

targets shows the current list of machines the machine health check is watching

Type

`array`

.status.targets[]

Type

`string`

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in MachineHealthCheck's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions represents the observations of a MachineHealthCheck's current state. Known condition types are RemediationAllowed, Paused.

.status.v1beta2.conditions

Description

conditions represents the observations of a MachineHealthCheck's current state. Known condition types are RemediationAllowed, Paused.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinehealthchecks`
  - `DELETE`: delete collection of MachineHealthCheck
  - `GET`: list objects of kind MachineHealthCheck
  - `POST`: create a new MachineHealthCheck
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinehealthchecks/{name}`
  - `DELETE`: delete the specified MachineHealthCheck
  - `GET`: read the specified MachineHealthCheck
  - `PATCH`: partially update the specified MachineHealthCheck
  - `PUT`: replace the specified MachineHealthCheck
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinehealthchecks/{name}/status`
  - `GET`: read status of the specified MachineHealthCheck
  - `PATCH`: partially update status of the specified MachineHealthCheck
  - `PUT`: replace status of the specified MachineHealthCheck

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinehealthchecks

HTTP method

`DELETE`

Description

delete collection of MachineHealthCheck

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind MachineHealthCheck

HTTP responses

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

HTTP method

`POST`

Description

create a new MachineHealthCheck

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`MachineHealthCheck` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`MachineHealthCheck` schema
201 - Created	`MachineHealthCheck` schema

HTTP code	Response body
202 - Accepted	`MachineHealthCheck` schema
401 - Unauthorized	Empty

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinehealthchecks/{name}

HTTP method

`DELETE`

Description

delete the specified MachineHealthCheck

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 MachineHealthCheck

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified MachineHealthCheck

Query parameters

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

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified MachineHealthCheck

Query parameters

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

Parameter	Type	Description
		request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`MachineHealthCheck` schema	`application/json` formatted

HTTP responses

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

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinehealthchecks/{name}/status

HTTP method

`GET`

Description

read status of the specified MachineHealthCheck

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified MachineHealthCheck

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified MachineHealthCheck

Query parameters

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`MachineHealthCheck` schema	`application/json` formatted

HTTP responses

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

MachineDrainRule [cluster.x-k8s.io/v1beta1]

Description

MachineDrainRule is the Schema for the MachineDrainRule API.

Type

`object`

Required

`metadata` `spec`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec defines the spec of a MachineDrainRule.

.spec

Description

spec defines the spec of a MachineDrainRule.

Type

`object`

Required

`drain`

Property	Type	Description
`drain`	`object`	drain configures if and how Pods are drained.

Property	Type	Description

`machines`	`array`	machines defines to which Machines this MachineDrainRule should be applied.
		
		If machines is not set, the MachineDrainRule applies to all Machines in the Namespace. If machines contains multiple selectors, the results are ORed. Within a single Machine selector the results of selector and clusterSelector are ANDed. Machines will be selected from all Clusters in the Namespace unless otherwise restricted with the clusterSelector.
		
		Example: Selects control plane Machines in all Clusters or Machines with label "os" == "linux" in Clusters with label "stage" == "production".
		
		• selector: matchExpressions:
		  • key: cluster.x-k8s.io/control-plane operator: Exists
		• selector: matchLabels: os: linux clusterSelector: matchExpressions:
		  • key: stage operator: In values:
		  • production

`pods`	`array`	pods defines to which Pods this MachineDrainRule should be applied. If pods is not set, the MachineDrainRule applies to all Pods in all Namespaces. If pods contains multiple selectors, the results are ORed. Within a single Pod selector the results of selector and namespaceSelector are ANDed. Pods will be selected from all Namespaces unless otherwise restricted with the namespaceSelector.

Property	Type	Description

		Example: Selects Pods with label "app" == "logging" in all Namespaces or Pods with label "app" == "prometheus" in the "monitoring" Namespace.
		• selector: matchExpressions:
		  • key: app operator: In values:
		  • logging
		• selector: matchLabels: app: prometheus namespaceSelector: matchLabels: kubernetes.io/metadata.name: monitoring

.spec.drain

Description

drain configures if and how Pods are drained.

Type

`object`

Required

`behavior`

Property	Type	Description
`behavior`	`string`	behavior defines the drain behavior. Can be either "Drain", "Skip", or "WaitCompleted". "Drain" means that the Pods to which this MachineDrainRule applies will be drained. If behavior is set to "Drain" the order in which Pods are drained can be configured with the order field. When draining Pods of a Node the Pods will be grouped by order and one group after another will be drained (by increasing order). Cluster API will wait until all Pods of a group are

Property	Type	Description
		terminated / removed from the Node before starting with the next group. "Skip" means that the Pods to which this MachineDrainRule applies will be skipped during drain. "WaitCompleted" means that the pods to which this MachineDrainRule applies will never be evicted and we wait for them to be completed, it is enforced that pods marked with this behavior always have Order=0.
`order`	`integer`	order defines the order in which Pods are drained. Pods with higher order are drained after Pods with lower order. order can only be set if behavior is set to "Drain". If order is not set, 0 will be used. Valid values for order are from -2147483648 to 2147483647 (inclusive).

.spec.machines

Description

machines defines to which Machines this MachineDrainRule should be applied. If machines is not set, the MachineDrainRule applies to all Machines in the Namespace. If machines contains multiple selectors, the results are ORed. Within a single Machine selector the results of selector and clusterSelector are ANDed. Machines will be selected from all Clusters in the Namespace unless otherwise restricted with the clusterSelector. Example: Selects control plane Machines in all Clusters or Machines with label "os" == "linux" in Clusters with label "stage" == "production". - selector: matchExpressions: - key: cluster.x-k8s.io/control-plane operator: Exists - selector: matchLabels: os: linux clusterSelector: matchExpressions: - key: stage operator: In values: - production

Type

`array`

.spec.machines[]

Description

MachineDrainRuleMachineSelector defines to which Machines this MachineDrainRule should be applied.

Type

`object`

Property	Type	Description
`clusterSelector`	`object`	clusterSelector is a label selector which selects Machines by the labels of their Clusters. This field follows standard label selector semantics; if not present or empty, it selects Machines of all Clusters. If selector is also set, then the selector as a whole selects Machines matching selector belonging to Clusters selected by clusterSelector. If selector is not set, it selects all Machines belonging to Clusters selected by clusterSelector.
`selector`	`object`	selector is a label selector which selects Machines by their labels. This field follows standard label selector semantics; if not present or empty, it selects all Machines. If clusterSelector is also set, then the selector as a whole selects Machines matching selector belonging to Clusters selected by clusterSelector. If clusterSelector is not set, it selects all Machines matching selector in all Clusters.

.spec.machines[].clusterSelector

Description

clusterSelector is a label selector which selects Machines by the labels of their Clusters. This field follows standard label selector semantics; if not present or empty, it selects Machines of all Clusters. If selector is also set, then the selector as a whole selects Machines matching selector belonging to Clusters selected by clusterSelector. If selector is not set, it selects all Machines belonging to Clusters selected by clusterSelector.

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.machines[].clusterSelector.matchExpressions

Description

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

Type

`array`

.spec.machines[].clusterSelector.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.machines[].clusterSelector.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.machines[].clusterSelector.matchExpressions[].values[]

Type

`string`

.spec.machines[].clusterSelector.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.machines[].selector

Description

selector is a label selector which selects Machines by their labels. This field follows standard label selector semantics; if not present or empty, it selects all Machines. If clusterSelector is also set, then the selector as a whole selects Machines matching selector belonging to Clusters selected by clusterSelector. If clusterSelector is not set, it selects all Machines matching selector in all Clusters.

Type

`object`

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

Property	Type	Description
		"key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.machines[].selector.matchExpressions

Description

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

Type

`array`

.spec.machines[].selector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

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

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

Type

`string`

.spec.machines[].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.pods

Description

pods defines to which Pods this MachineDrainRule should be applied. If pods is not set, the MachineDrainRule applies to all Pods in all Namespaces. If pods contains multiple selectors, the results are ORed. Within a single Pod selector the results of selector and namespaceSelector are ANDed. Pods will be selected from all Namespaces unless otherwise restricted with the namespaceSelector. Example: Selects Pods with label "app" == "logging" in all Namespaces or Pods with label "app" == "prometheus" in the "monitoring" Namespace. - selector: matchExpressions: - key: app operator: In values: - logging - selector: matchLabels: app: prometheus namespaceSelector: matchLabels: kubernetes.io/metadata.name: monitoring

Type

`array`

.spec.pods[]

Description

MachineDrainRulePodSelector defines to which Pods this MachineDrainRule should be applied.

Type

`object`

Property	Type	Description
`namespaceSelector`	`object`	namespaceSelector is a label selector which selects Pods by the labels of their Namespaces. This field follows standard label selector semantics; if not present or empty, it selects Pods of all Namespaces. If selector is also set, then the selector as a whole selects Pods matching selector in Namespaces selected by namespaceSelector. If selector is not set, it selects all Pods in Namespaces selected by namespaceSelector.

Property	Type	Description
`selector`	`object`	selector is a label selector which selects Pods by their labels. This field follows standard label selector semantics; if not present or empty, it selects all Pods. If namespaceSelector is also set, then the selector as a whole selects Pods matching selector in Namespaces selected by namespaceSelector. If namespaceSelector is not set, it selects all Pods matching selector in all Namespaces.

.spec.pods[].namespaceSelector

Description

namespaceSelector is a label selector which selects Pods by the labels of their Namespaces. This field follows standard label selector semantics; if not present or empty, it selects Pods of all Namespaces. If selector is also set, then the selector as a whole selects Pods matching selector in Namespaces selected by namespaceSelector. If selector is not set, it selects all Pods in Namespaces selected by namespaceSelector.

Type

`object`

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

Property	Type	Description
		"key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pods[].namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.pods[].namespaceSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

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

.spec.pods[].namespaceSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.pods[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.pods[].namespaceSelector.matchLabels

Description

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

Type

`object`

.spec.pods[].selector

Description

selector is a label selector which selects Pods by their labels. This field follows standard label selector semantics; if not present or empty, it selects all Pods. If namespaceSelector is also set, then the selector as a whole selects Pods matching selector in Namespaces selected by namespaceSelector. If namespaceSelector is not set, it selects all Pods matching selector in all Namespaces.

Type

`object`

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

.spec.pods[].selector.matchExpressions

Description

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

Type

`array`

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

Type

`string`

.spec.pods[].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`

API Endpoints

The following API endpoints are available:

- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedrainrules`
  - `DELETE`: delete collection of MachineDrainRule
  - `GET`: list objects of kind MachineDrainRule
  - `POST`: create a new MachineDrainRule
- `/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedrainrules/{name}`
  - `DELETE`: delete the specified MachineDrainRule
  - `GET`: read the specified MachineDrainRule
  - `PATCH`: partially update the specified MachineDrainRule
  - `PUT`: replace the specified MachineDrainRule

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedrainrules

HTTP method

`DELETE`

Description

delete collection of MachineDrainRule

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind MachineDrainRule

HTTP responses

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

HTTP method

`POST`

Description

create a new MachineDrainRule

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`MachineDrainRule` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`MachineDrainRule` schema
201 - Created	`MachineDrainRule` schema

HTTP code	Response body
202 - Accepted	`MachineDrainRule` schema
401 - Unauthorized	Empty

/apis/cluster.x-k8s.io/v1beta1/namespaces/{namespace}/machinedrainrules/{name}

HTTP method

`DELETE`

Description

delete the specified MachineDrainRule

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 MachineDrainRule

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified MachineDrainRule

Query parameters

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

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified MachineDrainRule

Query parameters

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

Parameter	Type	Description
		request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`MachineDrainRule` schema	`application/json` formatted

HTTP responses

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

Cluster API Add-on APIs

ClusterResourceSet ClusterResourceSetBinding

ClusterResourceSet [addons.cluster.x-k8s.io/v1beta1]

Description

ClusterResourceSet is the Schema for the clusterresourcesets API. For advanced use cases an add-on provider should be used instead.

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

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

.spec

Description

spec is the desired state of ClusterResourceSet.

Type

`object`

Required

`clusterSelector`

Property	Type	Description
`clusterSelector`	`object`	clusterSelector is the label selector for Clusters. The Clusters that are selected by this will be the ones affected by this ClusterResourceSet. It must match the Cluster labels. This field is immutable. Label selector cannot be empty.
`resources`	`array`	resources is a list of Secrets/ConfigMaps where each contains 1 or more resources to be applied to remote clusters.
`strategy`	`string`	strategy is the strategy to be used during applying resources. Defaults to ApplyOnce. This field is immutable.

.spec.clusterSelector

Description

clusterSelector is the label selector for Clusters. The Clusters that are selected by this will be the ones affected by this ClusterResourceSet. It must match the Cluster labels. This field is immutable. Label selector cannot be empty.

Type

`object`

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

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

.spec.clusterSelector.matchExpressions

Description

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

Type

`array`

.spec.clusterSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

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

.spec.clusterSelector.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.clusterSelector.matchExpressions[].values[]

Type

`string`

.spec.clusterSelector.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.resources

Description

resources is a list of Secrets/ConfigMaps where each contains 1 or more resources to be applied to remote clusters.

Type

`array`

.spec.resources[]

Description

ResourceRef specifies a resource.

Type

`object`

Required

`kind` `name`

Property	Type	Description
`kind`	`string`	kind of the resource. Supported kinds are: Secrets and ConfigMaps.
`name`	`string`	name of the resource that is in the same namespace with ClusterResourceSet object.

.status

Description

status is the observed state of ClusterResourceSet.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions defines current state of the ClusterResourceSet.
`observedGeneration`	`integer`	observedGeneration reflects the generation of the most recently observed ClusterResourceSet.
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in ClusterResourceSet's status with the V1Beta2 version.

.status.conditions

Description

conditions defines current state of the ClusterResourceSet.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in ClusterResourceSet's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions represents the observations of a ClusterResourceSet's current state. Known condition types are ResourceSetApplied, Deleting.

.status.v1beta2.conditions

Description

conditions represents the observations of a ClusterResourceSet's current state. Known condition types are ResourceSetApplied, Deleting.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last

Property	Type	Description
		transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesets`
  - `DELETE`: delete collection of ClusterResourceSet
  - `GET`: list objects of kind ClusterResourceSet
  - `POST`: create a new ClusterResourceSet
- `/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesets/{name}`
  - `DELETE`: delete the specified ClusterResourceSet
  - `GET`: read the specified ClusterResourceSet
  - `PATCH`: partially update the specified ClusterResourceSet
  - `PUT`: replace the specified ClusterResourceSet

- `/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesets/{name}/status`
  - `GET`: read status of the specified ClusterResourceSet
  - `PATCH`: partially update status of the specified ClusterResourceSet
  - `PUT`: replace status of the specified ClusterResourceSet

/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesets

HTTP method

`DELETE`

Description

delete collection of ClusterResourceSet

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind ClusterResourceSet

HTTP responses

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

HTTP method

`POST`

Description

create a new ClusterResourceSet

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`ClusterResourceSet` schema	`application/json` formatted

HTTP responses

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

/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesets/{name}

HTTP method

`DELETE`

Description

delete the specified ClusterResourceSet

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

HTTP code	Response body
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified ClusterResourceSet

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ClusterResourceSet

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

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.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified ClusterResourceSet

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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`	`ClusterResourceSet` schema	`application/json` formatted

HTTP responses

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

/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesets/{name}/status

HTTP method

`GET`

Description

read status of the specified ClusterResourceSet

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ClusterResourceSet

Query parameters

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

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified ClusterResourceSet

Query parameters

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

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ClusterResourceSet` schema	`application/json` formatted

HTTP responses

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

ClusterResourceSetBinding [addons.cluster.x-k8s.io/v1beta1]

Description

ClusterResourceSetBinding lists all matching ClusterResourceSets with the cluster it belongs to.

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

Property	Type	Description
		CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of ClusterResourceSetBinding.

.spec

Description

spec is the desired state of ClusterResourceSetBinding.

Type

`object`

Property	Type	Description
`bindings`	`array`	bindings is a list of ClusterResourceSets and their resources.
`clusterName`	`string`	clusterName is the name of the Cluster this binding applies to. Note: this field mandatory in v1beta2.

.spec.bindings

Description

bindings is a list of ClusterResourceSets and their resources.

Type

`array`

.spec.bindings[]

Description

ResourceSetBinding keeps info on all of the resources in a ClusterResourceSet.

Type

`object`

Required

`clusterResourceSetName`

Property	Type	Description
`clusterResourceSetName`	`string`	clusterResourceSetName is the name of the ClusterResourceSet that is applied to the owner cluster of the binding.
`resources`	`array`	resources is a list of resources that the ClusterResourceSet has.

.spec.bindings[].resources

Description

resources is a list of resources that the ClusterResourceSet has.

Type

`array`

.spec.bindings[].resources[]

Description

ResourceBinding shows the status of a resource that belongs to a ClusterResourceSet matched by the owner cluster of the ClusterResourceSetBinding object.

Type

`object`

Required

`applied` `kind` `name`

Property	Type	Description
`applied`	`boolean`	applied is to track if a resource is applied to the cluster or not.
`hash`	`string`	hash is the hash of a resource's data. This can be used to decide if a resource is changed. For "ApplyOnce" ClusterResourceSet.spec.strategy, this is no-op as that strategy does not act on change.
`kind`	`string`	kind of the resource. Supported kinds are: Secrets and ConfigMaps.
`lastAppliedTime`	`string`	lastAppliedTime identifies when this resource was last applied to the cluster.
`name`	`string`	name of the resource that is in the same namespace with ClusterResourceSet object.

API Endpoints

The following API endpoints are available:

- `/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesetbindings`
  - `DELETE`: delete collection of ClusterResourceSetBinding
  - `GET`: list objects of kind ClusterResourceSetBinding
  - `POST`: create a new ClusterResourceSetBinding
- `/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesetbindings/{name}`
  - `DELETE`: delete the specified ClusterResourceSetBinding
  - `GET`: read the specified ClusterResourceSetBinding
  - `PATCH`: partially update the specified ClusterResourceSetBinding
  - `PUT`: replace the specified ClusterResourceSetBinding
- `/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesetbindings/{name}/status`
  - `GET`: read status of the specified ClusterResourceSetBinding
  - `PATCH`: partially update status of the specified ClusterResourceSetBinding
  - `PUT`: replace status of the specified ClusterResourceSetBinding

/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesetbindings

HTTP method

`DELETE`

Description

delete collection of ClusterResourceSetBinding

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind ClusterResourceSetBinding

HTTP responses

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

HTTP method

`POST`

Description

create a new ClusterResourceSetBinding

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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`	`ClusterResourceSetBinding` schema	`application/json` formatted

HTTP responses

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

/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresource

setbindings/{name}

HTTP method

`DELETE`

Description

delete the specified ClusterResourceSetBinding

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 ClusterResourceSetBinding

HTTP responses

HTTP code	Response body
200 - OK	`ClusterResourceSetBinding` schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ClusterResourceSetBinding

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified ClusterResourceSetBinding

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`ClusterResourceSetBinding` schema	`application/json` formatted

HTTP responses

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

/apis/addons.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/clusterresourcesetbindings/{name}/status

HTTP method

`GET`

Description

read status of the specified ClusterResourceSetBinding

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ClusterResourceSetBinding

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified ClusterResourceSetBinding

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`ClusterResourceSetBinding` schema	`application/json` formatted

HTTP responses

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

Cluster API IPAM APIs

IPAddress

IPAddressClaim

IPAddress [ipam.cluster.x-k8s.io/v1alpha1]

Description

IPAddress is the Schema for the ipaddress API.

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of IPAddress.

.spec

Description

spec is the desired state of IPAddress.

Type

`object`

Required

`address` `claimRef` `poolRef` `prefix`

Property	Type	Description
`address`	`string`	address is the IP address.

Property	Type	Description
`claimRef`	`object`	claimRef is a reference to the claim this IPAddress was created for.
`gateway`	`string`	gateway is the network gateway of the network the address is from.
`poolRef`	`object`	poolRef is a reference to the pool that this IPAddress was created from.
`prefix`	`integer`	prefix is the prefix of the address.

.spec.claimRef

Description

claimRef is a reference to the claim this IPAddress was created for.

Type

`object`

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

.spec.poolRef

Description

poolRef is a reference to the pool that this IPAddress was created from.

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

API Endpoints

The following API endpoints are available:

- `/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses`
  - `DELETE`: delete collection of IPAddress
  - `GET`: list objects of kind IPAddress
  - `POST`: create a new IPAddress

- `/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses/{name}`
  - `DELETE`: delete the specified IPAddress
  - `GET`: read the specified IPAddress
  - `PATCH`: partially update the specified IPAddress
  - `PUT`: replace the specified IPAddress

/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses

HTTP method

`DELETE`

Description

delete collection of IPAddress

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind IPAddress

HTTP responses

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

HTTP method

`POST`

Description

create a new IPAddress

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`IPAddress` schema	`application/json` formatted

HTTP responses

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

/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses/{name}

HTTP method

`DELETE`

Description

delete the specified IPAddress

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

HTTP code	Response body
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified IPAddress

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified IPAddress

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

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.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified IPAddress

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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`	`IPAddress` schema	`application/json` formatted

HTTP responses

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

IPAddressClaim [ipam.cluster.x-k8s.io/v1alpha1]

Description

IPAddressClaim is the Schema for the ipaddressclaim 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:

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

.spec

Description

spec is the desired state of IPAddressClaim.

Type

`object`

Required

`poolRef`

Property	Type	Description
`clusterName`	`string`	clusterName is the name of the Cluster this object belongs to.

Property	Type	Description
`poolRef`	`object`	poolRef is a reference to the pool from which an IP address should be created.

.spec.poolRef

Description

poolRef is a reference to the pool from which an IP address should be created.

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

.status

Description

status is the observed state of IPAddressClaim.

Type

`object`

Property	Type	Description
`addressRef`	`object`	addressRef is a reference to the address that was created for this claim.
`conditions`	`array`	conditions summarises the current state of the IPAddressClaim
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in IPAddressClaim's status with the V1Beta2 version.

.status.addressRef

Description

addressRef is a reference to the address that was created for this claim.

Type

`object`

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

.status.conditions

Description

conditions summarises the current state of the IPAddressClaim

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed.

Property	Type	Description
		If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in IPAddressClaim's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions represents the observations of a IPAddressClaim's current state.

.status.v1beta2.conditions

Description

conditions represents the observations of a IPAddressClaim's current state.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last

Property	Type	Description
		transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims`
  - `DELETE`: delete collection of IPAddressClaim
  - `GET`: list objects of kind IPAddressClaim
  - `POST`: create a new IPAddressClaim
- `/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims/{name}`
  - `DELETE`: delete the specified IPAddressClaim
  - `GET`: read the specified IPAddressClaim
  - `PATCH`: partially update the specified IPAddressClaim
  - `PUT`: replace the specified IPAddressClaim

- `/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims/{name}/status`
  - `GET`: read status of the specified IPAddressClaim
  - `PATCH`: partially update status of the specified IPAddressClaim
  - `PUT`: replace status of the specified IPAddressClaim

/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims

HTTP method

`DELETE`

Description

delete collection of IPAddressClaim

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind IPAddressClaim

HTTP responses

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

HTTP method

`POST`

Description

create a new IPAddressClaim

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`IPAddressClaim` schema	`application/json` formatted

HTTP responses

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

/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims/{name}

HTTP method

`DELETE`

Description

delete the specified IPAddressClaim

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

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified IPAddressClaim

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified IPAddressClaim

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

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	`IPAddressClaim` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified IPAddressClaim

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently

Parameter	Type	Description
		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`	`IPAddressClaim` schema	`application/json` formatted

HTTP responses

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

/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims/{name}/status

HTTP method

`GET`

Description

read status of the specified IPAddressClaim

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified IPAddressClaim

Query parameters

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

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified IPAddressClaim

Query parameters

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

Parameter	Type	Description
		request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`IPAddressClaim` schema	`application/json` formatted

HTTP responses

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

Cluster API Runtime APIs

ExtensionConfig

ExtensionConfig [runtime.cluster.x-k8s.io/v1alpha1]

Description

ExtensionConfig is the Schema for the ExtensionConfig 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:

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

.spec

Description

spec is the desired state of the ExtensionConfig.

Type

`object`

Required

`clientConfig`

Property	Type	Description
`clientConfig`	`object`	clientConfig defines how to communicate with the Extension server.
`namespaceSelector`	`object`	namespaceSelector decides whether to call the hook for an object based on whether the

Property	Type	Description
		namespace for that object matches the selector. Defaults to the empty LabelSelector, which matches all objects.
`settings`	`object`	settings defines key value pairs to be passed to all calls to all supported RuntimeExtensions. Note: Settings can be overridden on the ClusterClass.

.spec.clientConfig

Description

clientConfig defines how to communicate with the Extension server.

Type

`object`

Property	Type	Description
`caBundle`	`string`	caBundle is a PEM encoded CA bundle which will be used to validate the Extension server's server certificate.
`service`	`object`	service is a reference to the Kubernetes service for the Extension server. Note: Exactly one of `url` or `service` must be specified. If the Extension server is running within a cluster, then you should use `service`.

Property	Type	Description
`url`	`string`	url gives the location of the Extension server, in standard URL form (`scheme://host:port/path`). Note: Exactly one of `url` or `service` must be specified. The scheme must be "https". The `host` should not refer to a service running in the cluster; use the `service` field instead. A path is optional, and if present may be any string permissible in a URL. If a path is set it will be used as prefix to the hook-specific path. Attempting to use a user or basic auth e.g. "user:password@" is not allowed. Fragments ("#...") and query parameters ("?...") are not allowed either.

.spec.clientConfig.service

Description

service is a reference to the Kubernetes service for the Extension server. Note: Exactly one of `url` or `service` must be specified. If the Extension server is running within a cluster, then you should use `service`.

Type

`object`

Required

`name` `namespace`

Property	Type	Description
`name`	`string`	name is the name of the service.
`namespace`	`string`	namespace is the namespace of the service.
`path`	`string`	path is an optional URL path and if present may be any string permissible in a URL. If a path is set it will be used as prefix to the hook-specific path.
`port`	`integer`	port is the port on the service that's hosting the Extension server. Defaults to 443. Port should be a valid port number (1-65535, inclusive).

.spec.namespaceSelector

Description

namespaceSelector decides whether to call the hook for an object based on whether the namespace for that object matches the selector. Defaults to the empty LabelSelector, which matches all objects.

Type

`object`

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

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

.spec.namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.namespaceSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

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

.spec.namespaceSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.namespaceSelector.matchLabels

Description

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

Type

`object`

.spec.settings

Description

settings defines key value pairs to be passed to all calls to all supported RuntimeExtensions. Note: Settings can be overridden on the ClusterClass.

Type

`object`

.status

Description

status is the current state of the ExtensionConfig

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions define the current service state of the ExtensionConfig.
`handlers`	`array`	handlers defines the current ExtensionHandlers supported by an Extension.
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in ExtensionConfig's status with the V1Beta2 version.

.status.conditions

Description

conditions define the current service state of the ExtensionConfig.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.

Property	Type	Description
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.handlers

Description

handlers defines the current ExtensionHandlers supported by an Extension.

Type

`array`

.status.handlers[]

Description

ExtensionHandler specifies the details of a handler for a particular runtime hook registered by an Extension server.

Type

`object`

Required

`name` `requestHook`

Property	Type	Description
`failurePolicy`	`string`	failurePolicy defines how failures in calls to the ExtensionHandler should be handled by a client. Defaults to Fail if not set.
`name`	`string`	name is the unique name of the ExtensionHandler.
`requestHook`	`object`	requestHook defines the versioned runtime hook which this ExtensionHandler serves.
`timeoutSeconds`	`integer`	timeoutSeconds defines the timeout duration for client calls to the ExtensionHandler. Defaults to 10 seconds if not set.

.status.handlers[].requestHook

Description

requestHook defines the versioned runtime hook which this ExtensionHandler serves.

Type

`object`

Required

`apiVersion` `hook`

Property	Type	Description
`apiVersion`	`string`	apiVersion is the group and version of the Hook.
`hook`	`string`	hook is the name of the hook.

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in ExtensionConfig's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions represents the observations of a ExtensionConfig's current state. Known condition types are Discovered, Paused.

.status.v1beta2.conditions

Description

conditions represents the observations of a ExtensionConfig's current state. Known condition types are Discovered, Paused.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.

Property	Type	Description
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/runtime.cluster.x-k8s.io/v1alpha1/extensionconfigs`
  - `DELETE`: delete collection of ExtensionConfig
  - `GET`: list objects of kind ExtensionConfig
  - `POST`: create a new ExtensionConfig
- `/apis/runtime.cluster.x-k8s.io/v1alpha1/extensionconfigs/{name}`
  - `DELETE`: delete the specified ExtensionConfig
  - `GET`: read the specified ExtensionConfig
  - `PATCH`: partially update the specified ExtensionConfig

- `PUT`: replace the specified ExtensionConfig

- `/apis/runtime.cluster.x-k8s.io/v1alpha1/extensionconfigs/{name}/status`
  - `GET`: read status of the specified ExtensionConfig
  - `PATCH`: partially update status of the specified ExtensionConfig
  - `PUT`: replace status of the specified ExtensionConfig

/apis/runtime.cluster.x-k8s.io/v1alpha1/extensionconfigs

HTTP method

`DELETE`

Description

delete collection of ExtensionConfig

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind ExtensionConfig

HTTP responses

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

HTTP method

`POST`

Description

create a new ExtensionConfig

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`ExtensionConfig` schema	`application/json` formatted

HTTP responses

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

/apis/runtime.cluster.x-k8s.io/v1alpha1/extensionconfigs/{name}

HTTP method

`DELETE`

Description

delete the specified ExtensionConfig

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 ExtensionConfig

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ExtensionConfig

Query parameters

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

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified ExtensionConfig

Query parameters

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

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ExtensionConfig` schema	`application/json` formatted

HTTP responses

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

/apis/runtime.cluster.x-k8s.io/v1alpha1/extensionconfigs/{name}/status

HTTP method

`GET`

Description

read status of the specified ExtensionConfig

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ExtensionConfig

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified ExtensionConfig

Query parameters

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ExtensionConfig` schema	`application/json` formatted

HTTP responses

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

Kubeadm Bootstrap APIs

KubeadmConfig

KubeadmConfigTemplate

KubeadmConfig [bootstrap.cluster.x-k8s.io/v1beta1]

Description

KubeadmConfig is the Schema for the kubeadmconfigs 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:

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

.spec

Description

spec is the desired state of KubeadmConfig.

Type

`object`

Property	Type	Description
`bootCommands`	`array`	bootCommands specifies extra commands to run very early in the boot process via the cloud-init bootcmd module. bootcmd will run on every boot, 'cloud-init-per' command can be used to make bootcmd run exactly once. This is typically run in the cloud-init.service systemd unit. This has no effect in Ignition.

Property	Type	Description
`clusterConfiguration`	`object`	clusterConfiguration along with InitConfiguration are the configurations necessary for the init command
`diskSetup`	`object`	diskSetup specifies options for the creation of partition tables and file systems on devices.
`files`	`array`	files specifies extra files to be passed to user_data upon creation.
`format`	`string`	format specifies the output format of the bootstrap data
`ignition`	`object`	ignition contains Ignition specific configuration.
`initConfiguration`	`object`	initConfiguration along with ClusterConfiguration are the configurations necessary for the init command
`joinConfiguration`	`object`	joinConfiguration is the kubeadm configuration for the join command

Property	Type	Description
`mounts`	`array`	mounts specifies a list of mount points to be setup.
`ntp`	`object`	ntp specifies NTP configuration
`postKubeadmCommands`	`array`	postKubeadmCommands specifies extra commands to run after kubeadm runs. With cloud-init, this is appended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is appended to /etc/kubeadm.sh.
`preKubeadmCommands`	`array`	preKubeadmCommands specifies extra commands to run before kubeadm runs. With cloud-init, this is prepended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is prepended to /etc/kubeadm.sh.
`useExperimentalRetryJoin`	`boolean`	useExperimentalRetryJoin replaces a basic kubeadm command with a shell script with retries for joins. This is meant to be an experimental temporary workaround on some environments where joins fail due to timing (and other issues). The long term goal is to

Property	Type	Description
		add retries to kubeadm proper and use that functionality. This will add about 40KB to userdata For more information, refer to https://github.com/kubernetes-sigs/cluster-api/pull/2763#discussion_r397306055↗. Deprecated: This experimental fix is no longer needed and this field will be removed in a future release. When removing also remove from staticcheck exclude-rules for SA1019 in golangci.yml
`users`	`array`	users specifies extra users to add
`verbosity`	`integer`	verbosity is the number for the kubeadm log level verbosity. It overrides the `--v` flag in kubeadm commands.

.spec.bootCommands

Description

bootCommands specifies extra commands to run very early in the boot process via the cloud-init bootcmd module. bootcmd will run on every boot, 'cloud-init-per' command can be used to make bootcmd run exactly once. This is typically run in the cloud-init.service systemd unit. This has no effect in Ignition.

Type

`array`

.spec.bootCommands[]

Type

`string`

.spec.clusterConfiguration

Description

clusterConfiguration along with InitConfiguration are the configurations necessary for the init command

Type

`object`

Property	Type	Description
`apiServer`	`object`	apiServer contains extra settings for the API server control plane component
`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 ↗
`certificatesDir`	`string`	certificatesDir specifies where to store or look for all required certificates. NB: if not provided, this will default to `/etc/kubernetes/pki`

Property	Type	Description
`clusterName`	`string`	clusterName is the cluster name
`controlPlaneEndpoint`	`string`	controlPlaneEndpoint sets a stable IP address or DNS name for the control plane; it can be a valid IP address or a RFC-1123 DNS subdomain, both with optional TCP port. In case the ControlPlaneEndpoint is not specified, the AdvertiseAddress + BindPort are used; in case the ControlPlaneEndpoint is specified but without a TCP port, the BindPort is used. Possible usages are: e.g. In a cluster with more than one control plane instances, this field should be assigned the address of the external load balancer in front of the control plane instances. e.g. in environments with enforced node recycling, the ControlPlaneEndpoint could be used for assigning a stable DNS to the control plane. NB: This value defaults to the first value in the Cluster object status.apiEndpoints array.
`controllerManager`	`object`	controllerManager contains extra settings for the controller manager control plane component
`dns`	`object`	dns defines the options for the DNS add-on installed in the cluster.

Property	Type	Description
`etcd`	`object`	etcd holds configuration for etcd. NB: This value defaults to a Local (stacked) etcd
`featureGates`	`object`	featureGates enabled by the user.

`imageRepository`	`string`	imageRepository sets the container registry to pull images from.
		• If not set, the default registry of kubeadm will be used, i.e.
		  • registry.k8s.io (new registry): >= v1.22.17, >= v1.23.15, >= v1.24.9, >= v1.25.0
		  • k8s.gcr.io (old registry): all older versions Please note that when imageRepository is not set we don't allow upgrades to versions >= v1.22.0 which use the old registry (k8s.gcr.io). Please use a newer patch version with the new registry instead (i.e. >= v1.22.17,
		> = v1.23.15, >= v1.24.9, >= v1.25.0).
		• If the version is a CI build (kubernetes version starts with `ci/` or `ci-cross/`) `gcr.io/k8s-staging-ci-images` will be used as a default for control plane components and for kube-proxy, while `registry.k8s.io` will be used for all the other images.

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`kubernetesVersion`	`string`	kubernetesVersion is the target version of the control plane. NB: This value defaults to the Machine object spec.version
`networking`	`object`	networking holds configuration for the networking topology of the cluster. NB: This value defaults to the Cluster object spec.clusterNetwork.
`scheduler`	`object`	scheduler contains extra settings for the scheduler control plane component

.spec.clusterConfiguration.apiServer

Description

apiServer contains extra settings for the API server control plane component

Type

`object`

Property	Type	Description
`certSANs`	`array`	certSANs sets extra Subject Alternative Names for the API Server signing cert.
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.
`timeoutForControlPlane`	`string`	timeoutForControlPlane controls the timeout that we use for API server to appear

.spec.clusterConfiguration.apiServer.certSANs

Description

certSANs sets extra Subject Alternative Names for the API Server signing cert.

Type

`array`

.spec.clusterConfiguration.apiServer.certSANs[]

Type

`string`

.spec.clusterConfiguration.apiServer.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.clusterConfiguration.apiServer.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.clusterConfiguration.apiServer.extraEnvs[]

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.clusterConfiguration.apiServer.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.clusterConfiguration.apiServer.extraEnvs[].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.clusterConfiguration.apiServer.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

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

.spec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.clusterConfiguration.apiServer.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.clusterConfiguration.apiServer.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.

Property	Type	Description
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.clusterConfiguration.controllerManager

Description

controllerManager contains extra settings for the controller manager control plane component

Type

`object`

Property	Type	Description
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.

.spec.clusterConfiguration.controllerManager.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.clusterConfiguration.controllerManager.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.clusterConfiguration.controllerManager.extraEnvs[]

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

Property	Type	Description
		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.clusterConfiguration.controllerManager.extraEnvs[].valueFrom

Description

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

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`fieldRef`	`object`	Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`,

Property	Type	Description
		spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
`secretKeyRef`	`object`	Selects a key of a secret in the pod's namespace

.spec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty.

Property	Type	Description
		Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.clusterConfiguration.controllerManager.extraEnvs[].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.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.clusterConfiguration.controllerManager.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.clusterConfiguration.controllerManager.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.clusterConfiguration.dns

Description

dns defines the options for the DNS add-on installed in the cluster.

Type

`object`

Property	Type	Description
`imageRepository`	`string`	imageRepository sets the container registry to pull images from. if not set, the ImageRepository defined in ClusterConfiguration will be used instead.
`imageTag`	`string`	imageTag allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.

.spec.clusterConfiguration.etcd

Description

etcd holds configuration for etcd. NB: This value defaults to a Local (stacked) etcd

Type

`object`

Property	Type	Description
`external`	`object`	external describes how to connect to an external etcd cluster Local and External are mutually exclusive
`local`	`object`	local provides configuration knobs for configuring the local etcd instance Local and External are mutually exclusive

.spec.clusterConfiguration.etcd.external

Description

external describes how to connect to an external etcd cluster Local and External are mutually exclusive

Type

`object`

Required

`caFile` `certFile` `endpoints` `keyFile`

Property	Type	Description
`caFile`	`string`	caFile is an SSL Certificate Authority file used to secure etcd communication. Required if using a TLS connection.
`certFile`	`string`	certFile is an SSL certification file used to secure etcd communication. Required if using a TLS connection.
`endpoints`	`array`	endpoints of etcd members. Required for ExternalEtcd.
`keyFile`	`string`	keyFile is an SSL key file used to secure etcd communication. Required if using a TLS connection.

.spec.clusterConfiguration.etcd.external.endpoints

Description

endpoints of etcd members. Required for ExternalEtcd.

Type

`array`

.spec.clusterConfiguration.etcd.external.endpoints[]

Type

`string`

.spec.clusterConfiguration.etcd.local

Description

local provides configuration knobs for configuring the local etcd instance Local and External are mutually exclusive

Type

`object`

Property	Type	Description
`dataDir`	`string`	dataDir is the directory etcd will place its data. Defaults to "/var/lib/etcd".
`extraArgs`	`object`	extraArgs are extra arguments provided to the etcd binary when run inside a static pod.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`imageRepository`	`string`	imageRepository sets the container registry to pull images from. if not set, the ImageRepository defined in ClusterConfiguration will be used instead.

Property	Type	Description
`imageTag`	`string`	imageTag allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.
`peerCertSANs`	`array`	peerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.
`serverCertSANs`	`array`	serverCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

.spec.clusterConfiguration.etcd.local.extraArgs

Description

extraArgs are extra arguments provided to the etcd binary when run inside a static pod.

Type

`object`

.spec.clusterConfiguration.etcd.local.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.clusterConfiguration.etcd.local.extraEnvs[]

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.clusterConfiguration.etcd.local.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.clusterConfiguration.etcd.local.extraEnvs[].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.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

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

.spec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.clusterConfiguration.etcd.local.peerCertSANs

Description

peerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.

Type

`array`

.spec.clusterConfiguration.etcd.local.peerCertSANs[]

Type

`string`

.spec.clusterConfiguration.etcd.local.serverCertSANs

Description

serverCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

Type

`array`

.spec.clusterConfiguration.etcd.local.serverCertSANs[]

Type

`string`

.spec.clusterConfiguration.featureGates

Description

featureGates enabled by the user.

Type

`object`

.spec.clusterConfiguration.networking

Description

networking holds configuration for the networking topology of the cluster. NB: This value defaults to the Cluster object spec.clusterNetwork.

Type

`object`

Property	Type	Description
`dnsDomain`	`string`	dnsDomain is the dns domain used by k8s services. Defaults to "cluster.local".
`podSubnet`	`string`	podSubnet is the subnet used by pods. If unset, the API server will not allocate CIDR ranges for every node. Defaults to a comma-delimited string of the Cluster object's spec.clusterNetwork.services.cidrBlocks if that is set
`serviceSubnet`	`string`	serviceSubnet is the subnet used by k8s services. Defaults to a comma-delimited string of the Cluster object's spec.clusterNetwork.pods.cidrBlocks, or to "10.96.0.0/12" if that's unset.

.spec.clusterConfiguration.scheduler

Description

scheduler contains extra settings for the scheduler control plane component

Type

`object`

Property	Type	Description
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.

.spec.clusterConfiguration.scheduler.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.clusterConfiguration.scheduler.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.clusterConfiguration.scheduler.extraEnvs[]

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.clusterConfiguration.scheduler.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.clusterConfiguration.scheduler.extraEnvs[].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.clusterConfiguration.scheduler.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key of the secret to select from. Must be a valid secret key.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost

Property	Type	Description
		certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.clusterConfiguration.scheduler.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.clusterConfiguration.scheduler.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.

Property	Type	Description
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.diskSetup

Description

diskSetup specifies options for the creation of partition tables and file systems on devices.

Type

`object`

Property	Type	Description
`filesystems`	`array`	filesystems specifies the list of file systems to setup.
`partitions`	`array`	partitions specifies the list of the partitions to setup.

.spec.diskSetup.filesystems

Description

filesystems specifies the list of file systems to setup.

Type

`array`

.spec.diskSetup.filesystems[]

Description

Filesystem defines the file systems to be created.

Type

`object`

Required

`device` `filesystem`

Property	Type	Description
`device`	`string`	device specifies the device name
`extraOpts`	`array`	extraOpts defined extra options to add to the command for creating the file system.
`filesystem`	`string`	filesystem specifies the file system type.
`label`	`string`	label specifies the file system label to be used. If set to None, no label is used.
`overwrite`	`boolean`	overwrite defines whether or not to overwrite any existing filesystem. If true, any pre-existing file system will be

Property	Type	Description
		destroyed. Use with Caution.
`partition`	`string`	partition specifies the partition to use. The valid options are: "auto|any", "auto", "any", "none", and , where NUM is the actual partition number.
`replaceFS`	`string`	replaceFS is a special directive, used for Microsoft Azure that instructs cloud-init to replace a file system of <FS_TYPE>. NOTE: unless you define a label, this requires the use of the 'any' partition directive.

.spec.diskSetup.filesystems[].extraOpts

Description

extraOpts defined extra options to add to the command for creating the file system.

Type

`array`

.spec.diskSetup.filesystems[].extraOpts[]

Type

`string`

.spec.diskSetup.partitions

Description

partitions specifies the list of the partitions to setup.

Type

`array`

.spec.diskSetup.partitions[]

Description

Partition defines how to create and layout a partition.

Type

`object`

Required

`device` `layout`

Property	Type	Description
`device`	`string`	device is the name of the device.
`layout`	`boolean`	layout specifies the device layout. If it is true, a single partition will be created for the entire device. When layout is false, it means don't partition or ignore existing partitioning.
`overwrite`	`boolean`	overwrite describes whether to skip checks and create the partition if a partition or filesystem is found on the device. Use with caution. Default is 'false'.
`tableType`	`string`	tableType specifies the tupe of partition table. The following are supported: 'mbr': default and setups a MS-DOS partition table 'gpt': setups a GPT partition table

.spec.files

Description

files specifies extra files to be passed to user_data upon creation.

Type

`array`

.spec.files[]

Description

File defines the input for generating write_files in cloud-init.

Type

`object`

Required

`path`

Property	Type	Description
`append`	`boolean`	append specifies whether to append Content to existing file if Path exists.
`content`	`string`	content is the actual content of the file.
`contentFrom`	`object`	contentFrom is a referenced source of content to populate the file.
`encoding`	`string`	encoding specifies the encoding of the file contents.
`owner`	`string`	owner specifies the ownership of the file, e.g. "root:root".

Property	Type	Description
`path`	`string`	path specifies the full path on disk where to store the file.
`permissions`	`string`	permissions specifies the permissions to assign to the file, e.g. "0640".

.spec.files[].contentFrom

Description

contentFrom is a referenced source of content to populate the file.

Type

`object`

Required

`secret`

Property	Type	Description
`secret`	`object`	secret represents a secret that should populate this file.

.spec.files[].contentFrom.secret

Description

secret represents a secret that should populate this file.

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	key is the key in the secret's data map for this value.
`name`	`string`	name of the secret in the KubeadmBootstrapConfig's namespace to use.

.spec.ignition

Description

ignition contains Ignition specific configuration.

Type

`object`

Property	Type	Description
`containerLinuxConfig`	`object`	containerLinuxConfig contains CLC specific configuration.

.spec.ignition.containerLinuxConfig

Description

containerLinuxConfig contains CLC specific configuration.

Type

`object`

Property	Type	Description
`additionalConfig`	`string`	additionalConfig contains additional configuration to be merged with the Ignition configuration generated by the bootstrapper controller. More info: https://coreos.github.io/ignition/operator-notes/#config-merging ↗ The data format is documented here: https://kinvolk.io/docs/flatcar-container-linux/latest/provisioning/cl-config/ ↗
`strict`	`boolean`	strict controls if AdditionalConfig should be strictly parsed. If so, warnings are treated as errors.

.spec.initConfiguration

Description

initConfiguration along with ClusterConfiguration are the configurations necessary for the init command

Type

`object`

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`bootstrapTokens`	`array`	bootstrapTokens is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature
`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 ↗
`localAPIEndpoint`	`object`	localAPIEndpoint represents the endpoint of the API server instance that's deployed on this control plane node In HA setups, this differs from ClusterConfiguration.ControlPlaneEndpoint in the sense that ControlPlaneEndpoint is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.
`nodeRegistration`	`object`	nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes,

Property	Type	Description
		NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration
`patches`	`object`	patches contains options related to applying patches to components deployed by kubeadm during "kubeadm init". The minimum kubernetes version needed to support Patches is v1.22
`skipPhases`	`array`	skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

.spec.initConfiguration.bootstrapTokens

Description

bootstrapTokens is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature

Type

`array`

.spec.initConfiguration.bootstrapTokens[]

Description

BootstrapToken describes one bootstrap token, stored as a Secret in the cluster.

Type

`object`

Required

`token`

Property	Type	Description
`description`	`string`	description sets a human-friendly message why this token exists and what it's used for, so other administrators can know its purpose.
`expires`	`string`	expires specifies the timestamp when this token expires. Defaults to being set dynamically at runtime based on the TTL. Expires and TTL are mutually exclusive.
`groups`	`array`	groups specifies the extra groups that this token will authenticate as when/if used for authentication
`token`	`string`	token is used for establishing bidirectional trust between nodes and control-planes. Used for joining nodes in the cluster.
`ttl`	`string`	ttl defines the time to live for this token. Defaults to 24h. Expires and TTL are mutually exclusive.
`usages`	`array`	usages describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.

.spec.initConfiguration.bootstrapTokens[].groups

Description

groups specifies the extra groups that this token will authenticate as when/if used for authentication

Type

`array`

.spec.initConfiguration.bootstrapTokens[].groups[]

Type

`string`

.spec.initConfiguration.bootstrapTokens[].usages

Description

usages describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.

Type

`array`

.spec.initConfiguration.bootstrapTokens[].usages[]

Type

`string`

.spec.initConfiguration.localAPIEndpoint

Description

localAPIEndpoint represents the endpoint of the API server instance that's deployed on this control plane node In HA setups, this differs from ClusterConfiguration.ControlPlaneEndpoint in the sense that ControlPlaneEndpoint is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of

the default interface and use that, but in case that process fails you may set the desired value here.

Type

`object`

Property	Type	Description
`advertiseAddress`	`string`	advertiseAddress sets the IP address for the API server to advertise.
`bindPort`	`integer`	bindPort sets the secure port for the API Server to bind to. Defaults to 6443.

.spec.initConfiguration.nodeRegistration

Description

nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Type

`object`

Property	Type	Description
`criSocket`	`string`	criSocket is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use
`ignorePreflightErrors`	`array`	ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current

Property	Type	Description
		node is registered.
`imagePullPolicy`	`string`	imagePullPolicy specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". Defaults to "IfNotPresent". This can be used only with Kubernetes version equal to 1.22 and later.
`imagePullSerial`	`boolean`	imagePullSerial specifies if image pulling performed by kubeadm must be done serially or in parallel. This option takes effect only on Kubernetes >=1.31.0. Default: true (defaulted in kubeadm)
`kubeletExtraArgs`	`object`	kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.
`name`	`string`	name is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the CommonName

Property	Type	Description
		field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`taints`	`array`	taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

.spec.initConfiguration.nodeRegistration.ignorePreflightErrors

Description

ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Type

`array`

.spec.initConfiguration.nodeRegistration.ignorePreflightErrors[]

Type

`string`

.spec.initConfiguration.nodeRegistration.kubeletExtraArgs

Description

kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.

Type

`object`

.spec.initConfiguration.nodeRegistration.taints

Description

taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""'}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

Type

`array`

.spec.initConfiguration.nodeRegistration.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.initConfiguration.patches

Description

patches contains options related to applying patches to components deployed by kubeadm during "kubeadm init". The minimum kubernetes version needed to support Patches is v1.22

Type

`object`

Property	Type	Description
`directory`	`string`	directory is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic"

Property	Type	Description
		"merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically. These files can be written into the target directory via KubeadmConfig.Files which specifies additional files to be created on the machine, either with content inline or by referencing a secret.

.spec.initConfiguration.skipPhases

Description

skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

Type

`array`

.spec.initConfiguration.skipPhases[]

Type

`string`

.spec.joinConfiguration

Description

joinConfiguration is the kubeadm configuration for the join command

Type

`object`

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 ↗
`caCertPath`	`string`	caCertPath is the path to the SSL certificate authority used to secure comunications between node and control-plane. Defaults to "/etc/kubernetes/pki/ca.crt".
`controlPlane`	`object`	controlPlane defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.
`discovery`	`object`	discovery specifies the options for the kubelet to use during the TLS Bootstrap process
`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 ↗
`nodeRegistration`	`object`	nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used

Property	Type	Description
		in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration
`patches`	`object`	patches contains options related to applying patches to components deployed by kubeadm during "kubeadm join". The minimum kubernetes version needed to support Patches is v1.22
`skipPhases`	`array`	skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

.spec.joinConfiguration.controlPlane

Description

controlPlane defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.

Type

`object`

Property	Type	Description
`localAPIEndpoint`	`object`	localAPIEndpoint represents the endpoint of the API server instance to be deployed on this node.

.spec.joinConfiguration.controlPlane.localAPIEndpoint

Description

localAPIEndpoint represents the endpoint of the API server instance to be deployed on this node.

Type

`object`

Property	Type	Description
`advertiseAddress`	`string`	advertiseAddress sets the IP address for the API server to advertise.
`bindPort`	`integer`	bindPort sets the secure port for the API Server to bind to. Defaults to 6443.

.spec.joinConfiguration.discovery

Description

discovery specifies the options for the kubelet to use during the TLS Bootstrap process

Type

`object`

Property	Type	Description
`bootstrapToken`	`object`	bootstrapToken is used to set the options for bootstrap token based discovery BootstrapToken and File are mutually exclusive

Property	Type	Description
`file`	`object`	file is used to specify a file or URL to a kubeconfig file from which to load cluster information BootstrapToken and File are mutually exclusive
`timeout`	`string`	timeout modifies the discovery timeout
`tlsBootstrapToken`	`string`	tlsBootstrapToken is a token used for TLS bootstrapping. If .BootstrapToken is set, this field is defaulted to .BootstrapToken.Token, but can be overridden. If .File is set, this field **must be set** in case the KubeConfigFile does not contain any other authentication information

.spec.joinConfiguration.discovery.bootstrapToken

Description

bootstrapToken is used to set the options for bootstrap token based discovery BootstrapToken and File are mutually exclusive

Type

`object`

Property	Type	Description
`apiServerEndpoint`	`string`	apiServerEndpoint is an IP or domain name to the API server from which info will be fetched.

Property	Type	Description
`caCertHashes`	`array`	caCertHashes specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as ":", where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL: openssl x509 -pubkey -in ca.crt openssl rsa -pubin -outform der 2>&/dev/null | openssl dgst -sha256 -hex
`token`	`string`	token is a token used to validate cluster information fetched from the control-plane.
`unsafeSkipCAVerification`	`boolean`	unsafeSkipCAVerification allows token-based discovery without CA verification via CACertHashes. This can weaken the security of kubeadm since other nodes can impersonate the control-plane.

.spec.joinConfiguration.discovery.bootstrapToken.caCertHashes

Description

caCertHashes specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as "<type>:<value>", where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL: openssl x509 -pubkey -in ca.crt openssl rsa -pubin -outform der 2>&/dev/null | openssl dgst -sha256 -hex

Type

`array`

.spec.joinConfiguration.discovery.bootstrapToken.caCertHashes[]

Type

`string`

.spec.joinConfiguration.discovery.file

Description

file is used to specify a file or URL to a kubeconfig file from which to load cluster information BootstrapToken and File are mutually exclusive

Type

`object`

Required

`kubeConfigPath`

Property	Type	Description
`kubeConfig`	`object`	kubeConfig is used (optionally) to generate a KubeConfig based on the KubeadmConfig's information. The file is generated at the path specified in KubeConfigPath.

Property	Type	Description
		Host address (server field) information is automatically populated based on the Cluster's ControlPlaneEndpoint. Certificate Authority (certificate-authority-data field) is gathered from the cluster's CA secret.
`kubeConfigPath`	`string`	kubeConfigPath is used to specify the actual file path or URL to the kubeconfig file from which to load cluster information

.spec.joinConfiguration.discovery.file.kubeConfig

Description

kubeConfig is used (optionally) to generate a KubeConfig based on the KubeadmConfig's information. The file is generated at the path specified in KubeConfigPath. Host address (server field) information is automatically populated based on the Cluster's ControlPlaneEndpoint. Certificate Authority (certificate-authority-data field) is gathered from the cluster's CA secret.

Type

`object`

Required

`user`

Property	Type	Description
`cluster`	`object`	cluster contains information about how to communicate with the kubernetes cluster. By default the following fields are automatically populated: • Server with the Cluster's ControlPlaneEndpoint.

Property	Type	Description
		• CertificateAuthorityData with the Cluster's CA certificate.
`user`	`object`	user contains information that describes identity information. This is used to tell the kubernetes cluster who you are.

.spec.joinConfiguration.discovery.file.kubeConfig.cluster

Description

cluster contains information about how to communicate with the kubernetes cluster. By default the following fields are automatically populated: - Server with the Cluster's ControlPlaneEndpoint. - CertificateAuthorityData with the Cluster's CA certificate.

Type

`object`

Property	Type	Description
`certificateAuthorityData`	`string`	certificateAuthorityData contains PEM-encoded certificate authority certificates. Defaults to the Cluster's CA certificate if empty.
`insecureSkipTLSVerify`	`boolean`	insecureSkipTLSVerify skips the validity check for the server's certificate. This will make your HTTPS connections insecure.

Property	Type	Description
`proxyURL`	`string`	proxyURL is the URL to the proxy to be used for all requests made by this client. URLs with "http", "https", and "socks5" schemes are supported. If this configuration is not provided or the empty string, the client attempts to construct a proxy configuration from http_proxy and https_proxy environment variables. If these environment variables are not set, the client does not attempt to proxy requests. socks5 proxying does not currently support spdy streaming endpoints (exec, attach, port forward).
`server`	`string`	server is the address of the kubernetes cluster (https://hostname:port↗). Defaults to https:// + Cluster.Spec.ControlPlaneEndpoint.
`tlsServerName`	`string`	tlsServerName is used to check server certificate. If TLSServerName is empty, the hostname used to contact the server is used.

.spec.joinConfiguration.discovery.file.kubeConfig.user

Description

user contains information that describes identity information. This is used to tell the kubernetes cluster who you are.

Type

`object`

Property	Type	Description
`authProvider`	`object`	authProvider specifies a custom authentication plugin for the kubernetes cluster.
`exec`	`object`	exec specifies a custom exec-based authentication plugin for the kubernetes cluster.

.spec.joinConfiguration.discovery.file.kubeConfig.user.authProvider

Description

authProvider specifies a custom authentication plugin for the kubernetes cluster.

Type

`object`

Required

`name`

Property	Type	Description
`config`	`object`	config holds the parameters for the authentication plugin.
`name`	`string`	name is the name of the authentication plugin.

.spec.joinConfiguration.discovery.file.kubeConfig.user.authProvider.config

Description

config holds the parameters for the authentication plugin.

Type

`object`

.spec.joinConfiguration.discovery.file.kubeConfig.user.exec

Description

exec specifies a custom exec-based authentication plugin for the kubernetes cluster.

Type

`object`

Required

`command`

Property	Type	Description
`apiVersion`	`string`	apiVersion is preferred input version of the ExecInfo. The returned ExecCredentials MUST use the same encoding version as the input. Defaults to client.authentication.k8s.io/v1 if not set.
`args`	`array`	args is the arguments to pass to the command when executing it.
`command`	`string`	command to execute.

Property	Type	Description
`env`	`array`	env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.
`provideClusterInfo`	`boolean`	provideClusterInfo determines whether or not to provide cluster information, which could potentially contain very large CA data, to this exec plugin as a part of the KUBERNETES_EXEC_INFO environment variable. By default, it is set to false. Package k8s.io/client-go/tools/auth/exec provides helper methods for reading this environment variable.

.spec.joinConfiguration.discovery.file.kubeConfig.user.exec.args

Description

args is the arguments to pass to the command when executing it.

Type

`array`

.spec.joinConfiguration.discovery.file.kubeConfig.user.exec.args[]

Type

`string`

.spec.joinConfiguration.discovery.file.kubeConfig.user.exec.env

Description

env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

Type

`array`

.spec.joinConfiguration.discovery.file.kubeConfig.user.exec.env[]

Description

KubeConfigAuthExecEnv is used for setting environment variables when executing an exec-based credential plugin.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	name of the environment variable
`value`	`string`	value of the environment variable

.spec.joinConfiguration.nodeRegistration

Description

nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Type

`object`

Property	Type	Description
`criSocket`	`string`	criSocket is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use
`ignorePreflightErrors`	`array`	ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.
`imagePullPolicy`	`string`	imagePullPolicy specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". Defaults to "IfNotPresent". This can be used only with Kubernetes version equal to 1.22 and later.
`imagePullSerial`	`boolean`	imagePullSerial specifies if image pulling performed by kubeadm must be done serially or in parallel. This option takes effect only on Kubernetes >=1.31.0. Default: true (defaulted in kubeadm)

Property	Type	Description
`kubeletExtraArgs`	`object`	kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.
`name`	`string`	name is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the CommonName field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`taints`	`array`	taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

.spec.joinConfiguration.nodeRegistration.ignorePreflightErrors

Description

ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Type

`array`

.spec.joinConfiguration.nodeRegistration.ignorePreflightErrors[]

Type

`string`

.spec.joinConfiguration.nodeRegistration.kubeletExtraArgs

Description

kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.

Type

`object`

.spec.joinConfiguration.nodeRegistration.taints

Description

taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-

role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

Type

`array`

.spec.joinConfiguration.nodeRegistration.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.joinConfiguration.patches

Description

patches contains options related to applying patches to components deployed by kubeadm during "kubeadm join". The minimum kubernetes version needed to support Patches is v1.22

Type

`object`

Property	Type	Description
`directory`	`string`	directory is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic" "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically. These files can be written into the target directory via KubeadmConfig.Files which specifies additional files to be created on the machine, either with content inline or by referencing a secret.

.spec.joinConfiguration.skipPhases

Description

skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

Type

`array`

.spec.joinConfiguration.skipPhases[]

Type

`string`

.spec.mounts

Description

mounts specifies a list of mount points to be setup.

Type

`array`

.spec.mounts[]

Description

MountPoints defines input for generated mounts in cloud-init.

Type

`array`

.spec.ntp

Description

ntp specifies NTP configuration

Type

`object`

Property	Type	Description
`enabled`	`boolean`	enabled specifies whether NTP should be enabled

Property	Type	Description
`servers`	`array`	servers specifies which NTP servers to use

.spec.ntp.servers

Description

servers specifies which NTP servers to use

Type

`array`

.spec.ntp.servers[]

Type

`string`

.spec.postKubeadmCommands

Description

postKubeadmCommands specifies extra commands to run after kubeadm runs. With cloud-init, this is appended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is appended to /etc/kubeadm.sh.

Type

`array`

.spec.postKubeadmCommands[]

Type

`string`

.spec.preKubeadmCommands

Description

preKubeadmCommands specifies extra commands to run before kubeadm runs. With cloud-init, this is prepended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is prepended to /etc/kubeadm.sh.

Type

`array`

.spec.preKubeadmCommands[]

Type

`string`

.spec.users

Description

users specifies extra users to add

Type

`array`

.spec.users[]

Description

User defines the input for a generated user in cloud-init.

Type

`object`

Required

`name`

Property	Type	Description
`gecos`	`string`	gecos specifies the gecos to use for the user
`groups`	`string`	groups specifies the additional groups for the user
`homeDir`	`string`	homeDir specifies the home directory to use for the user
`inactive`	`boolean`	inactive specifies whether to mark the user as inactive
`lockPassword`	`boolean`	lockPassword specifies if password login should be disabled
`name`	`string`	name specifies the user name
`passwd`	`string`	passwd specifies a hashed password for the user
`passwdFrom`	`object`	passwdFrom is a referenced source of passwd to populate the passwd.
`primaryGroup`	`string`	primaryGroup specifies the primary group for the user

Property	Type	Description
`shell`	`string`	shell specifies the user's shell
`sshAuthorizedKeys`	`array`	sshAuthorizedKeys specifies a list of ssh authorized keys for the user
`sudo`	`string`	sudo specifies a sudo role for the user

.spec.users[].passwdFrom

Description

passwdFrom is a referenced source of passwd to populate the passwd.

Type

`object`

Required

`secret`

Property	Type	Description
`secret`	`object`	secret represents a secret that should populate this password.

.spec.users[].passwdFrom.secret

Description

secret represents a secret that should populate this password.

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	key is the key in the secret's data map for this value.
`name`	`string`	name of the secret in the KubeadmBootstrapConfig's namespace to use.

.spec.users[].sshAuthorizedKeys

Description

sshAuthorizedKeys specifies a list of ssh authorized keys for the user

Type

`array`

.spec.users[].sshAuthorizedKeys[]

Type

`string`

.status

Description

status is the observed state of KubeadmConfig.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions defines current service state of the KubeadmConfig.
`dataSecretName`	`string`	dataSecretName is the name of the secret that stores the bootstrap data script.
`failureMessage`	`string`	failureMessage will be set on non-retryable errors Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`failureReason`	`string`	failureReason will be set on non-retryable errors Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`observedGeneration`	`integer`	observedGeneration is the latest generation observed by the controller.
`ready`	`boolean`	ready indicates the BootstrapData field is ready to be consumed

Property	Type	Description
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in KubeadmConfig's status with the V1Beta2 version.

.status.conditions

Description

conditions defines current service state of the KubeadmConfig.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.

Property	Type	Description
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in KubeadmConfig's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions represents the observations of a KubeadmConfig's current state. Known condition types are Ready, DataSecretAvailable, CertificatesAvailable.

.status.v1beta2.conditions

Description

conditions represents the observations of a KubeadmConfig's current state. Known condition types are Ready, DataSecretAvailable, CertificatesAvailable.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigs`
  - `DELETE`: delete collection of KubeadmConfig
  - `GET`: list objects of kind KubeadmConfig
  - `POST`: create a new KubeadmConfig
- `/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigs/{name}`
  - `DELETE`: delete the specified KubeadmConfig
  - `GET`: read the specified KubeadmConfig
  - `PATCH`: partially update the specified KubeadmConfig
  - `PUT`: replace the specified KubeadmConfig
- `/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigs/{name}/status`
  - `GET`: read status of the specified KubeadmConfig
  - `PATCH`: partially update status of the specified KubeadmConfig

- `PUT`: replace status of the specified KubeadmConfig

/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigs

HTTP method

`DELETE`

Description

delete collection of KubeadmConfig

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind KubeadmConfig

HTTP responses

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

HTTP method

`POST`

Description

create a new KubeadmConfig

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`KubeadmConfig` schema	`application/json` formatted

HTTP responses

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

/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigs/{name}

HTTP method

`DELETE`

Description

delete the specified KubeadmConfig

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

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified KubeadmConfig

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified KubeadmConfig

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

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	`KubeadmConfig` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified KubeadmConfig

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently

Parameter	Type	Description
		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`	`KubeadmConfig` schema	`application/json` formatted

HTTP responses

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

/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigs/{name}/status

HTTP method

`GET`

Description

read status of the specified KubeadmConfig

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified KubeadmConfig

Query parameters

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

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified KubeadmConfig

Query parameters

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

Parameter	Type	Description
		request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`KubeadmConfig` schema	`application/json` formatted

HTTP responses

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

KubeadmConfigTemplate [bootstrap.cluster.x-k8s.io/v1beta1]

Description

KubeadmConfigTemplate is the Schema for the kubeadmconfigtemplates 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:

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

.spec

Description

spec is the desired state of KubeadmConfigTemplate.

Type

`object`

Required

`template`

Property	Type	Description
`template`	`object`	template defines the desired state of KubeadmConfigTemplate.

.spec.template

Description

template defines the desired state of KubeadmConfigTemplate.

Type

`object`

Property	Type	Description
`metadata`	`ObjectMeta`	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`spec`	`object`	spec is the desired state of KubeadmConfig.

.spec.template.spec

Description

spec is the desired state of KubeadmConfig.

Type

`object`

Property	Type	Description
`bootCommands`	`array`	bootCommands specifies extra commands to run very early in the boot process via the cloud-init bootcmd module. bootcmd will run on every boot, 'cloud-init-per' command can be used to make bootcmd run exactly once. This is typically run in the cloud-init.service systemd unit. This has no effect in Ignition.

Property	Type	Description
`clusterConfiguration`	`object`	clusterConfiguration along with InitConfiguration are the configurations necessary for the init command
`diskSetup`	`object`	diskSetup specifies options for the creation of partition tables and file systems on devices.
`files`	`array`	files specifies extra files to be passed to user_data upon creation.
`format`	`string`	format specifies the output format of the bootstrap data
`ignition`	`object`	ignition contains Ignition specific configuration.
`initConfiguration`	`object`	initConfiguration along with ClusterConfiguration are the configurations necessary for the init command
`joinConfiguration`	`object`	joinConfiguration is the kubeadm configuration for the join command

Property	Type	Description
`mounts`	`array`	mounts specifies a list of mount points to be setup.
`ntp`	`object`	ntp specifies NTP configuration
`postKubeadmCommands`	`array`	postKubeadmCommands specifies extra commands to run after kubeadm runs. With cloud-init, this is appended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is appended to /etc/kubeadm.sh.
`preKubeadmCommands`	`array`	preKubeadmCommands specifies extra commands to run before kubeadm runs. With cloud-init, this is prepended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is prepended to /etc/kubeadm.sh.
`useExperimentalRetryJoin`	`boolean`	useExperimentalRetryJoin replaces a basic kubeadm command with a shell script with retries for joins. This is meant to be an experimental temporary workaround on some environments where joins fail due to timing (and other issues). The long term goal is to

Property	Type	Description
		add retries to kubeadm proper and use that functionality. This will add about 40KB to userdata For more information, refer to https://github.com/kubernetes-sigs/cluster-api/pull/2763#discussion_r397306055↗. Deprecated: This experimental fix is no longer needed and this field will be removed in a future release. When removing also remove from staticcheck exclude-rules for SA1019 in golangci.yml
`users`	`array`	users specifies extra users to add
`verbosity`	`integer`	verbosity is the number for the kubeadm log level verbosity. It overrides the `--v` flag in kubeadm commands.

.spec.template.spec.bootCommands

Description

bootCommands specifies extra commands to run very early in the boot process via the cloud-init bootcmd module. bootcmd will run on every boot, 'cloud-init-per' command can be used to make bootcmd run exactly once. This is typically run in the cloud-init.service systemd unit. This has no effect in Ignition.

Type

`array`

.spec.template.spec.bootCommands[]

Type

`string`

.spec.template.spec.clusterConfiguration

Description

clusterConfiguration along with InitConfiguration are the configurations necessary for the init command

Type

`object`

Property	Type	Description
`apiServer`	`object`	apiServer contains extra settings for the API server control plane component
`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 ↗
`certificatesDir`	`string`	certificatesDir specifies where to store or look for all required certificates. NB: if not provided, this will default to `/etc/kubernetes/pki`

Property	Type	Description
`clusterName`	`string`	clusterName is the cluster name
`controlPlaneEndpoint`	`string`	controlPlaneEndpoint sets a stable IP address or DNS name for the control plane; it can be a valid IP address or a RFC-1123 DNS subdomain, both with optional TCP port. In case the ControlPlaneEndpoint is not specified, the AdvertiseAddress + BindPort are used; in case the ControlPlaneEndpoint is specified but without a TCP port, the BindPort is used. Possible usages are: e.g. In a cluster with more than one control plane instances, this field should be assigned the address of the external load balancer in front of the control plane instances. e.g. in environments with enforced node recycling, the ControlPlaneEndpoint could be used for assigning a stable DNS to the control plane. NB: This value defaults to the first value in the Cluster object status.apiEndpoints array.
`controllerManager`	`object`	controllerManager contains extra settings for the controller manager control plane component
`dns`	`object`	dns defines the options for the DNS add-on installed in the cluster.

Property	Type	Description
`etcd`	`object`	etcd holds configuration for etcd. NB: This value defaults to a Local (stacked) etcd
`featureGates`	`object`	featureGates enabled by the user.

`imageRepository`	`string`	imageRepository sets the container registry to pull images from.
		• If not set, the default registry of kubeadm will be used, i.e.
		  • registry.k8s.io (new registry): >= v1.22.17, >= v1.23.15, >= v1.24.9, >= v1.25.0
		  • k8s.gcr.io (old registry): all older versions Please note that when imageRepository is not set we don't allow upgrades to versions >= v1.22.0 which use the old registry (k8s.gcr.io). Please use a newer patch version with the new registry instead (i.e. >= v1.22.17,
		> = v1.23.15, >= v1.24.9, >= v1.25.0).
		• If the version is a CI build (kubernetes version starts with `ci/` or `ci-cross/`) `gcr.io/k8s-staging-ci-images` will be used as a default for control plane components and for kube-proxy, while `registry.k8s.io` will be used for all the other images.

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`kubernetesVersion`	`string`	kubernetesVersion is the target version of the control plane. NB: This value defaults to the Machine object spec.version
`networking`	`object`	networking holds configuration for the networking topology of the cluster. NB: This value defaults to the Cluster object spec.clusterNetwork.
`scheduler`	`object`	scheduler contains extra settings for the scheduler control plane component

.spec.template.spec.clusterConfiguration.apiServer

Description

apiServer contains extra settings for the API server control plane component

Type

`object`

Property	Type	Description
`certSANs`	`array`	certSANs sets extra Subject Alternative Names for the API Server signing cert.
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.
`timeoutForControlPlane`	`string`	timeoutForControlPlane controls the timeout that we use for API server to appear

.spec.template.spec.clusterConfiguration.apiServer.certSANs

Description

certSANs sets extra Subject Alternative Names for the API Server signing cert.

Type

`array`

.spec.template.spec.clusterConfiguration.apiServer.certSANs[]

Type

`string`

.spec.template.spec.clusterConfiguration.apiServer.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.template.spec.clusterConfiguration.apiServer.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.template.spec.clusterConfiguration.apiServer.extraEnvs[]

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.template.spec.clusterConfiguration.apiServer.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.template.spec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.template.spec.clusterConfiguration.apiServer.extraEnvs[].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.template.spec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

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

.spec.template.spec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.template.spec.clusterConfiguration.apiServer.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.template.spec.clusterConfiguration.apiServer.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.

Property	Type	Description
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.template.spec.clusterConfiguration.controllerManager

Description

controllerManager contains extra settings for the controller manager control plane component

Type

`object`

Property	Type	Description
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Property	Type	Description
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.

.spec.template.spec.clusterConfiguration.controllerManager.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.template.spec.clusterConfiguration.controllerManager.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.template.spec.clusterConfiguration.controllerManager.extraEnvs[]

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.template.spec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.template.spec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.template.spec.clusterConfiguration.controllerManager.extraEnvs[].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.template.spec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

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

.spec.template.spec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.template.spec.clusterConfiguration.controllerManager.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.template.spec.clusterConfiguration.controllerManager.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.

Property	Type	Description
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.template.spec.clusterConfiguration.dns

Description

dns defines the options for the DNS add-on installed in the cluster.

Type

`object`

Property	Type	Description
`imageRepository`	`string`	imageRepository sets the container registry to pull images from. if not set, the ImageRepository defined in ClusterConfiguration will be used instead.
`imageTag`	`string`	imageTag allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.

.spec.template.spec.clusterConfiguration.etcd

Description

etcd holds configuration for etcd. NB: This value defaults to a Local (stacked) etcd

Type

`object`

Property	Type	Description
`external`	`object`	external describes how to connect to an external etcd cluster Local and External are mutually exclusive
`local`	`object`	local provides configuration knobs for configuring the local etcd instance Local and External are mutually exclusive

.spec.template.spec.clusterConfiguration.etcd.external

Description

external describes how to connect to an external etcd cluster Local and External are mutually exclusive

Type

`object`

Required

`caFile` `certFile` `endpoints` `keyFile`

Property	Type	Description
`caFile`	`string`	caFile is an SSL Certificate Authority file used to secure etcd communication. Required if using a TLS connection.
`certFile`	`string`	certFile is an SSL certification file used to secure etcd communication. Required if using a TLS connection.

Property	Type	Description
`endpoints`	`array`	endpoints of etcd members. Required for ExternalEtcd.
`keyFile`	`string`	keyFile is an SSL key file used to secure etcd communication. Required if using a TLS connection.

.spec.template.spec.clusterConfiguration.etcd.external.endpoints

Description

endpoints of etcd members. Required for ExternalEtcd.

Type

`array`

.spec.template.spec.clusterConfiguration.etcd.external.endpoints[]

Type

`string`

.spec.template.spec.clusterConfiguration.etcd.local

Description

local provides configuration knobs for configuring the local etcd instance Local and External are mutually exclusive

Type

`object`

Property	Type	Description
`dataDir`	`string`	dataDir is the directory etcd will place its data. Defaults to "/var/lib/etcd".
`extraArgs`	`object`	extraArgs are extra arguments provided to the etcd binary when run inside a static pod.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`imageRepository`	`string`	imageRepository sets the container registry to pull images from. if not set, the ImageRepository defined in ClusterConfiguration will be used instead.
`imageTag`	`string`	imageTag allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.
`peerCertSANs`	`array`	peerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.

Property	Type	Description
`serverCertSANs`	`array`	serverCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

.spec.template.spec.clusterConfiguration.etcd.local.extraArgs

Description

extraArgs are extra arguments provided to the etcd binary when run inside a static pod.

Type

`object`

.spec.template.spec.clusterConfiguration.etcd.local.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.template.spec.clusterConfiguration.etcd.local.extraEnvs[]

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.template.spec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.template.spec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.template.spec.clusterConfiguration.etcd.local.extraEnvs[].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.template.spec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

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

.spec.template.spec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.template.spec.clusterConfiguration.etcd.local.peerCertSANs

Description

peerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.

Type

`array`

.spec.template.spec.clusterConfiguration.etcd.local.peerCertSANs[]

Type

`string`

.spec.template.spec.clusterConfiguration.etcd.local.serverCertSANs

Description

serverCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

Type

`array`

.spec.template.spec.clusterConfiguration.etcd.local.serverCertSANs[]

Type

`string`

.spec.template.spec.clusterConfiguration.featureGates

Description

featureGates enabled by the user.

Type

`object`

.spec.template.spec.clusterConfiguration.networking

Description

networking holds configuration for the networking topology of the cluster. NB: This value defaults to the Cluster object spec.clusterNetwork.

Type

`object`

Property	Type	Description
`dnsDomain`	`string`	dnsDomain is the dns domain used by k8s services. Defaults to "cluster.local".
`podSubnet`	`string`	podSubnet is the subnet used by pods. If unset, the API server will not allocate CIDR ranges for every node. Defaults to a comma-delimited string of the Cluster object's spec.clusterNetwork.services.cidrBlocks if that is set
`serviceSubnet`	`string`	serviceSubnet is the subnet used by k8s services. Defaults to a comma-delimited string of the Cluster object's spec.clusterNetwork.pods.cidrBlocks, or to "10.96.0.0/12" if that's unset.

.spec.template.spec.clusterConfiguration.scheduler

Description

scheduler contains extra settings for the scheduler control plane component

Type

`object`

Property	Type	Description
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.

.spec.template.spec.clusterConfiguration.scheduler.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.template.spec.clusterConfiguration.scheduler.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.template.spec.clusterConfiguration.scheduler.extraEnvs[]

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	Name of the environment variable. Must be a C_IDENTIFIER.
`value`	`string`	Variable references $(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double $$ are reduced to a

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

.spec.template.spec.clusterConfiguration.scheduler.extraEnvs[].valueFrom

Description

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

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`fieldRef`	`object`	Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
`secretKeyRef`	`object`	Selects a key of a secret in the pod's namespace

.spec.template.spec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.template.spec.clusterConfiguration.scheduler.extraEnvs[].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.template.spec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.template.spec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.template.spec.clusterConfiguration.scheduler.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.template.spec.clusterConfiguration.scheduler.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.template.spec.diskSetup

Description

diskSetup specifies options for the creation of partition tables and file systems on devices.

Type

`object`

Property	Type	Description
`filesystems`	`array`	filesystems specifies the list of file systems to setup.

Property	Type	Description
`partitions`	`array`	partitions specifies the list of the partitions to setup.

.spec.template.spec.diskSetup.filesystems

Description

filesystems specifies the list of file systems to setup.

Type

`array`

.spec.template.spec.diskSetup.filesystems[]

Description

Filesystem defines the file systems to be created.

Type

`object`

Required

`device` `filesystem`

Property	Type	Description
`device`	`string`	device specifies the device name
`extraOpts`	`array`	extraOpts defined extra options to add to the command for creating the file system.

Property	Type	Description
`filesystem`	`string`	filesystem specifies the file system type.
`label`	`string`	label specifies the file system label to be used. If set to None, no label is used.
`overwrite`	`boolean`	overwrite defines whether or not to overwrite any existing filesystem. If true, any pre-existing file system will be destroyed. Use with Caution.
`partition`	`string`	partition specifies the partition to use. The valid options are: "auto|any", "auto", "any", "none", and , where NUM is the actual partition number.
`replaceFS`	`string`	replaceFS is a special directive, used for Microsoft Azure that instructs cloud-init to replace a file system of <FS_TYPE>. NOTE: unless you define a label, this requires the use of the 'any' partition directive.

.spec.template.spec.diskSetup.filesystems[].extraOpts

Description

extraOpts defined extra options to add to the command for creating the file system.

Type

`array`

.spec.template.spec.diskSetup.filesystems[].extraOpts[]

Type

`string`

.spec.template.spec.diskSetup.partitions

Description

partitions specifies the list of the partitions to setup.

Type

`array`

.spec.template.spec.diskSetup.partitions[]

Description

Partition defines how to create and layout a partition.

Type

`object`

Required

`device` `layout`

Property	Type	Description
`device`	`string`	device is the name of the device.
`layout`	`boolean`	layout specifies the device layout. If it is true, a single partition will be created for the entire device. When layout is false, it means don't partition or ignore existing partitioning.
`overwrite`	`boolean`	overwrite describes whether to skip checks and create the partition if a partition or filesystem is found on the device.

Property	Type	Description
		Use with caution. Default is 'false'.
`tableType`	`string`	tableType specifies the tupe of partition table. The following are supported: 'mbr': default and setups a MS-DOS partition table 'gpt': setups a GPT partition table

.spec.template.spec.files

Description

files specifies extra files to be passed to user_data upon creation.

Type

`array`

.spec.template.spec.files[]

Description

File defines the input for generating write_files in cloud-init.

Type

`object`

Required

`path`

Property	Type	Description
`append`	`boolean`	append specifies whether to append Content to existing file if Path exists.

Property	Type	Description
`content`	`string`	content is the actual content of the file.
`contentFrom`	`object`	contentFrom is a referenced source of content to populate the file.
`encoding`	`string`	encoding specifies the encoding of the file contents.
`owner`	`string`	owner specifies the ownership of the file, e.g. "root:root".
`path`	`string`	path specifies the full path on disk where to store the file.
`permissions`	`string`	permissions specifies the permissions to assign to the file, e.g. "0640".

.spec.template.spec.files[].contentFrom

Description

contentFrom is a referenced source of content to populate the file.

Type

`object`

Required

`secret`

Property	Type	Description
`secret`	`object`	secret represents a secret that should populate this file.

.spec.template.spec.files[].contentFrom.secret

Description

secret represents a secret that should populate this file.

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	key is the key in the secret's data map for this value.
`name`	`string`	name of the secret in the KubeadmBootstrapConfig's namespace to use.

.spec.template.spec.ignition

Description

ignition contains Ignition specific configuration.

Type

`object`

Property	Type	Description
`containerLinuxConfig`	`object`	containerLinuxConfig contains CLC specific configuration.

.spec.template.spec.ignition.containerLinuxConfig

Description

containerLinuxConfig contains CLC specific configuration.

Type

`object`

Property	Type	Description
`additionalConfig`	`string`	additionalConfig contains additional configuration to be merged with the Ignition configuration generated by the bootstrapper controller. More info: https://coreos.github.io/ignition/operator-notes/#config-merging ↗ The data format is documented here: https://kinvolk.io/docs/flatcar-container-linux/latest/provisioning/cl-config/ ↗
`strict`	`boolean`	strict controls if AdditionalConfig should be strictly parsed. If so, warnings are treated as errors.

.spec.template.spec.initConfiguration

Description

initConfiguration along with ClusterConfiguration are the configurations necessary for the init command

Type

`object`

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 ↗
`bootstrapTokens`	`array`	bootstrapTokens is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature
`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 ↗
`localAPIEndpoint`	`object`	localAPIEndpoint represents the endpoint of the API server instance that's deployed on this control plane node In HA setups, this differs from ClusterConfiguration.ControlPlaneEndpoint in the

Property	Type	Description
		sense that ControlPlaneEndpoint is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.
`nodeRegistration`	`object`	nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration
`patches`	`object`	patches contains options related to applying patches to components deployed by kubeadm during "kubeadm init". The minimum kubernetes version needed to support Patches is v1.22
`skipPhases`	`array`	skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

.spec.template.spec.initConfiguration.bootstrapTokens

Description

bootstrapTokens is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature

Type

`array`

.spec.template.spec.initConfiguration.bootstrapTokens[]

Description

BootstrapToken describes one bootstrap token, stored as a Secret in the cluster.

Type

`object`

Required

`token`

Property	Type	Description
`description`	`string`	description sets a human-friendly message why this token exists and what it's used for, so other administrators can know its purpose.
`expires`	`string`	expires specifies the timestamp when this token expires. Defaults to being set dynamically at runtime based on the TTL. Expires and TTL are mutually exclusive.
`groups`	`array`	groups specifies the extra groups that this token will authenticate as when/if used for authentication
`token`	`string`	token is used for establishing bidirectional trust between nodes and control-planes. Used for joining nodes in the

Property	Type	Description
		cluster.
`ttl`	`string`	ttl defines the time to live for this token. Defaults to 24h. Expires and TTL are mutually exclusive.
`usages`	`array`	usages describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.

.spec.template.spec.initConfiguration.bootstrapTokens[].groups

Description

groups specifies the extra groups that this token will authenticate as when/if used for authentication

Type

`array`

.spec.template.spec.initConfiguration.bootstrapTokens[].groups[]

Type

`string`

.spec.template.spec.initConfiguration.bootstrapTokens[].usages

Description

usages describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.

Type

`array`

.spec.template.spec.initConfiguration.bootstrapTokens[].usages[]

Type

`string`

.spec.template.spec.initConfiguration.localAPIEndpoint

Description

localAPIEndpoint represents the endpoint of the API server instance that's deployed on this control plane node In HA setups, this differs from ClusterConfiguration.ControlPlaneEndpoint in the sense that ControlPlaneEndpoint is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.

Type

`object`

Property	Type	Description
`advertiseAddress`	`string`	advertiseAddress sets the IP address for the API server to advertise.

Property	Type	Description
`bindPort`	`integer`	bindPort sets the secure port for the API Server to bind to. Defaults to 6443.

.spec.template.spec.initConfiguration.nodeRegistration

Description

nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Type

`object`

Property	Type	Description
`criSocket`	`string`	criSocket is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use
`ignorePreflightErrors`	`array`	ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Property	Type	Description
`imagePullPolicy`	`string`	imagePullPolicy specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". Defaults to "IfNotPresent". This can be used only with Kubernetes version equal to 1.22 and later.
`imagePullSerial`	`boolean`	imagePullSerial specifies if image pulling performed by kubeadm must be done serially or in parallel. This option takes effect only on Kubernetes >=1.31.0. Default: true (defaulted in kubeadm)
`kubeletExtraArgs`	`object`	kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.
`name`	`string`	name is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the CommonName field of the kubelet's client certificate to the API

Property	Type	Description
		server. Defaults to the hostname of the node if not provided.
`taints`	`array`	taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

.spec.template.spec.initConfiguration.nodeRegistration.ignorePreflightErrors

Description

ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Type

`array`

.spec.template.spec.initConfiguration.nodeRegistration.ignorePreflightErrors[]

Type

`string`

.spec.template.spec.initConfiguration.nodeRegistration.kubeletExtraArgs

Description

kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.

Type

`object`

.spec.template.spec.initConfiguration.nodeRegistration.taints

Description

taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""'}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

Type

`array`

.spec.template.spec.initConfiguration.nodeRegistration.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.template.spec.initConfiguration.patches

Description

patches contains options related to applying patches to components deployed by kubeadm during "kubeadm init". The minimum kubernetes version needed to support Patches is v1.22

Type

`object`

Property	Type	Description
`directory`	`string`	directory is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-

Property	Type	Description
		apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic" "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically. These files can be written into the target directory via KubeadmConfig.Files which specifies additional files to be created on the machine, either with content inline or by referencing a secret.

.spec.template.spec.initConfiguration.skipPhases

Description

skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

Type

`array`

.spec.template.spec.initConfiguration.skipPhases[]

Type

`string`

.spec.template.spec.joinConfiguration

Description

joinConfiguration is the kubeadm configuration for the join command

Type

`object`

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 ↗
`caCertPath`	`string`	caCertPath is the path to the SSL certificate authority used to secure comunications between node and control-plane. Defaults to "/etc/kubernetes/pki/ca.crt".
`controlPlane`	`object`	controlPlane defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.
`discovery`	`object`	discovery specifies the options for the kubelet to use during the TLS Bootstrap process
`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
`nodeRegistration`	`object`	nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration
`patches`	`object`	patches contains options related to applying patches to components deployed by kubeadm during "kubeadm join". The minimum kubernetes version needed to support Patches is v1.22
`skipPhases`	`array`	skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

.spec.template.spec.joinConfiguration.controlPlane

Description

controlPlane defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.

Type

`object`

Property	Type	Description
`localAPIEndpoint`	`object`	localAPIEndpoint represents the endpoint of the API server instance to be deployed on this node.

.spec.template.spec.joinConfiguration.controlPlane.localAPIEndpoint

Description

localAPIEndpoint represents the endpoint of the API server instance to be deployed on this node.

Type

`object`

Property	Type	Description
`advertiseAddress`	`string`	advertiseAddress sets the IP address for the API server to advertise.
`bindPort`	`integer`	bindPort sets the secure port for the API Server to bind to. Defaults to 6443.

.spec.template.spec.joinConfiguration.discovery

Description

discovery specifies the options for the kubelet to use during the TLS Bootstrap process

Type

`object`

Property	Type	Description
`bootstrapToken`	`object`	bootstrapToken is used to set the options for bootstrap token based discovery BootstrapToken and File are mutually exclusive
`file`	`object`	file is used to specify a file or URL to a kubeconfig file from which to load cluster information BootstrapToken and File are mutually exclusive
`timeout`	`string`	timeout modifies the discovery timeout
`tlsBootstrapToken`	`string`	tlsBootstrapToken is a token used for TLS bootstrapping. If .BootstrapToken is set, this field is defaulted to .BootstrapToken.Token, but can be overridden. If .File is set, this field **must be set** in case the KubeConfigFile does not contain any other authentication information

.spec.template.spec.joinConfiguration.discovery.bootstrapToken

Description

bootstrapToken is used to set the options for bootstrap token based discovery BootstrapToken and File are mutually exclusive

Type

`object`

Property	Type	Description
`apiServerEndpoint`	`string`	apiServerEndpoint is an IP or domain name to the API server from which info will be fetched.
`caCertHashes`	`array`	caCertHashes specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as ":", where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL: openssl x509 -pubkey -in ca.crt openssl rsa -pubin -outform der 2>&/dev/null | openssl dgst -sha256 -hex
`token`	`string`	token is a token used to validate cluster information fetched from the control-plane.
`unsafeSkipCAVerification`	`boolean`	unsafeSkipCAVerification allows token-based discovery without CA verification via CACertHashes. This can weaken the security of kubeadm since other nodes can impersonate the control-plane.

.spec.template.spec.joinConfiguration.discovery.bootstrapToken.caCertHashes

Description

caCertHashes specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as "<type>:<value>", where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL: openssl x509 -pubkey -in ca.crt openssl rsa -pubin -outform der 2>&/dev/null | openssl dgst -sha256 -hex

Type

`array`

.spec.template.spec.joinConfiguration.discovery.bootstrapToken.caCertHashes[]

Type

`string`

.spec.template.spec.joinConfiguration.discovery.file

Description

file is used to specify a file or URL to a kubeconfig file from which to load cluster information BootstrapToken and File are mutually exclusive

Type

`object`

Required

`kubeConfigPath`

Property	Type	Description
`kubeConfig`	`object`	kubeConfig is used (optionally) to generate a KubeConfig based on the KubeadmConfig's information. The file is generated at the path specified in KubeConfigPath. Host address (server field) information is automatically populated based on the Cluster's ControlPlaneEndpoint. Certificate Authority (certificate-authority-data field) is gathered from the cluster's CA secret.
`kubeConfigPath`	`string`	kubeConfigPath is used to specify the actual file path or URL to the kubeconfig file from which to load cluster information

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig

Description

kubeConfig is used (optionally) to generate a KubeConfig based on the KubeadmConfig's information. The file is generated at the path specified in KubeConfigPath. Host address (server field) information is automatically populated based on the Cluster's ControlPlaneEndpoint. Certificate Authority (certificate-authority-data field) is gathered from the cluster's CA secret.

Type

`object`

Required

`user`

Property	Type	Description
`cluster`	`object`	cluster contains information about how to communicate with the kubernetes cluster. By default the following fields are automatically populated: • Server with the Cluster's ControlPlaneEndpoint. • CertificateAuthorityData with the Cluster's CA certificate.
`user`	`object`	user contains information that describes identity information. This is used to tell the kubernetes cluster who you are.

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.cluster

Description

cluster contains information about how to communicate with the kubernetes cluster. By default the following fields are automatically populated: - Server with the Cluster's ControlPlaneEndpoint. - CertificateAuthorityData with the Cluster's CA certificate.

Type

`object`

Property	Type	Description
`certificateAuthorityData`	`string`	certificateAuthorityData contains PEM-encoded certificate authority certificates. Defaults to the Cluster's CA certificate if empty.
`insecureSkipTLSVerify`	`boolean`	insecureSkipTLSVerify skips the validity check for the server's certificate. This will make your HTTPS connections insecure.
`proxyURL`	`string`	proxyURL is the URL to the proxy to be used for all requests made by this client. URLs with "http", "https", and "socks5" schemes are supported. If this configuration is not provided or the empty string, the client attempts to construct a proxy configuration from http_proxy and https_proxy environment variables. If these environment variables are not set, the client does not attempt to proxy requests. socks5 proxying does not currently support spdy streaming endpoints (exec, attach, port forward).
`server`	`string`	server is the address of the kubernetes cluster (https://hostname:port ↗). Defaults to https:// + Cluster.Spec.ControlPlaneEndpoint.

Property	Type	Description
`tlsServerName`	`string`	tlsServerName is used to check server certificate. If TLSServerName is empty, the hostname used to contact the server is used.

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.user

Description

user contains information that describes identity information. This is used to tell the kubernetes cluster who you are.

Type

`object`

Property	Type	Description
`authProvider`	`object`	authProvider specifies a custom authentication plugin for the kubernetes cluster.
`exec`	`object`	exec specifies a custom exec-based authentication plugin for the kubernetes cluster.

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.user.authProvider

Description

authProvider specifies a custom authentication plugin for the kubernetes cluster.

Type

`object`

Required

`name`

Property	Type	Description
`config`	`object`	config holds the parameters for the authentication plugin.
`name`	`string`	name is the name of the authentication plugin.

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.user.authProvider.config

Description

config holds the parameters for the authentication plugin.

Type

`object`

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.user.exec

Description

exec specifies a custom exec-based authentication plugin for the kubernetes cluster.

Type

`object`

Required

`command`

Property	Type	Description
`apiVersion`	`string`	apiVersion is preferred input version of the ExecInfo. The returned ExecCredentials MUST use the same encoding version as the input. Defaults to client.authentication.k8s.io/v1 if not set.
`args`	`array`	args is the arguments to pass to the command when executing it.
`command`	`string`	command to execute.
`env`	`array`	env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.
`provideClusterInfo`	`boolean`	provideClusterInfo determines whether or not to provide cluster information, which could potentially contain very large CA data, to this exec plugin as a part of the KUBERNETES_EXEC_INFO environment variable. By default, it is set to false. Package k8s.io/client-go/tools/auth/exec provides helper methods for reading this environment variable.

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.user.exec.args

Description

args is the arguments to pass to the command when executing it.

Type

`array`

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.user.exec.args[]

Type

`string`

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.user.exec.env

Description

env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

Type

`array`

.spec.template.spec.joinConfiguration.discovery.file.kubeConfig.user.exec.env[]

Description

KubeConfigAuthExecEnv is used for setting environment variables when executing an exec-based credential plugin.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	name of the environment variable
`value`	`string`	value of the environment variable

.spec.template.spec.joinConfiguration.nodeRegistration

Description

nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Type

`object`

Property	Type	Description
`criSocket`	`string`	criSocket is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use
`ignorePreflightErrors`	`array`	ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.
`imagePullPolicy`	`string`	imagePullPolicy specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". Defaults

Property	Type	Description
		to "IfNotPresent". This can be used only with Kubernetes version equal to 1.22 and later.
`imagePullSerial`	`boolean`	imagePullSerial specifies if image pulling performed by kubeadm must be done serially or in parallel. This option takes effect only on Kubernetes >=1.31.0. Default: true (defaulted in kubeadm)
`kubeletExtraArgs`	`object`	kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.
`name`	`string`	name is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the CommonName field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`taints`	`array`	taints specifies the taints the Node API object should be registered with. If this field is unset,

Property	Type	Description
		i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

.spec.template.spec.joinConfiguration.nodeRegistration.ignorePreflightErrors

Description

ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Type

`array`

.spec.template.spec.joinConfiguration.nodeRegistration.ignorePreflightErrors[]

Type

`string`

.spec.template.spec.joinConfiguration.nodeRegistration.kubeletExtraArgs

Description

kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-

config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.

Type

`object`

.spec.template.spec.joinConfiguration.nodeRegistration.taints

Description

taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""'}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

Type

`array`

.spec.template.spec.joinConfiguration.nodeRegistration.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule,

Property	Type	Description
		PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.template.spec.joinConfiguration.patches

Description

patches contains options related to applying patches to components deployed by kubeadm during "kubeadm join". The minimum kubernetes version needed to support Patches is v1.22

Type

`object`

Property	Type	Description
`directory`	`string`	directory is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic" "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an

Property	Type	Description
		optional string that can be used to determine which patches are applied first alpha-numerically. These files can be written into the target directory via KubeadmConfig.Files which specifies additional files to be created on the machine, either with content inline or by referencing a secret.

.spec.template.spec.joinConfiguration.skipPhases

Description

skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

Type

`array`

.spec.template.spec.joinConfiguration.skipPhases[]

Type

`string`

.spec.template.spec.mounts

Description

mounts specifies a list of mount points to be setup.

Type

`array`

.spec.template.spec.mounts[]

Description

MountPoints defines input for generated mounts in cloud-init.

Type

`array`

.spec.template.spec.ntp

Description

ntp specifies NTP configuration

Type

`object`

Property	Type	Description
`enabled`	`boolean`	enabled specifies whether NTP should be enabled
`servers`	`array`	servers specifies which NTP servers to use

.spec.template.spec.ntp.servers

Description

servers specifies which NTP servers to use

Type

`array`

.spec.template.spec.ntp.servers[]

Type

`string`

.spec.template.spec.postKubeadmCommands

Description

postKubeadmCommands specifies extra commands to run after kubeadm runs. With cloud-init, this is appended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is appended to /etc/kubeadm.sh.

Type

`array`

.spec.template.spec.postKubeadmCommands[]

Type

`string`

.spec.template.spec.preKubeadmCommands

Description

preKubeadmCommands specifies extra commands to run before kubeadm runs. With cloud-init, this is prepended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is prepended to /etc/kubeadm.sh.

Type

`array`

.spec.template.spec.preKubeadmCommands[]

Type

`string`

.spec.template.spec.users

Description

users specifies extra users to add

Type

`array`

.spec.template.spec.users[]

Description

User defines the input for a generated user in cloud-init.

Type

`object`

Required

`name`

Property	Type	Description
`gecos`	`string`	gecos specifies the gecos to use for the user
`groups`	`string`	groups specifies the additional groups for the user
`homeDir`	`string`	homeDir specifies the home directory to use for the user
`inactive`	`boolean`	inactive specifies whether to mark the user as inactive
`lockPassword`	`boolean`	lockPassword specifies if password login should be disabled

Property	Type	Description
`name`	`string`	name specifies the user name
`passwd`	`string`	passwd specifies a hashed password for the user
`passwdFrom`	`object`	passwdFrom is a referenced source of passwd to populate the passwd.
`primaryGroup`	`string`	primaryGroup specifies the primary group for the user
`shell`	`string`	shell specifies the user's shell
`sshAuthorizedKeys`	`array`	sshAuthorizedKeys specifies a list of ssh authorized keys for the user
`sudo`	`string`	sudo specifies a sudo role for the user

.spec.template.spec.users[].passwdFrom

Description

passwdFrom is a referenced source of passwd to populate the passwd.

Type

`object`

Required

`secret`

Property	Type	Description
`secret`	`object`	secret represents a secret that should populate this password.

.spec.template.spec.users[].passwdFrom.secret

Description

secret represents a secret that should populate this password.

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	key is the key in the secret's data map for this value.
`name`	`string`	name of the secret in the KubeadmBootstrapConfig's namespace to use.

.spec.template.spec.users[].sshAuthorizedKeys

Description

sshAuthorizedKeys specifies a list of ssh authorized keys for the user

Type

`array`

.spec.template.spec.users[].sshAuthorizedKeys[]

Type

`string`

API Endpoints

The following API endpoints are available:

- `/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigtemplates`
  - `DELETE`: delete collection of KubeadmConfigTemplate
  - `GET`: list objects of kind KubeadmConfigTemplate
  - `POST`: create a new KubeadmConfigTemplate
- `/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigtemplates/{name}`
  - `DELETE`: delete the specified KubeadmConfigTemplate
  - `GET`: read the specified KubeadmConfigTemplate
  - `PATCH`: partially update the specified KubeadmConfigTemplate
  - `PUT`: replace the specified KubeadmConfigTemplate

/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigtemplates

HTTP method

`DELETE`

Description

delete collection of KubeadmConfigTemplate

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind KubeadmConfigTemplate

HTTP responses

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

HTTP method

`POST`

Description

create a new KubeadmConfigTemplate

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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`	`KubeadmConfigTemplate` schema	`application/json` formatted

HTTP responses

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

/apis/bootstrap.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmconfigt

emplates/{name}

HTTP method

`DELETE`

Description

delete the specified KubeadmConfigTemplate

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 KubeadmConfigTemplate

HTTP responses

HTTP code	Response body
200 - OK	`KubeadmConfigTemplate` schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified KubeadmConfigTemplate

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified KubeadmConfigTemplate

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`KubeadmConfigTemplate` schema	`application/json` formatted

HTTP responses

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

Kubeadm Control Plane APIs

KubeadmControlPlane KubeadmControlPlaneTemplate

KubeadmControlPlane [controlplane.cluster.x-k8s.io/v1beta1]

Description

KubeadmControlPlane is the Schema for the KubeadmControlPlane 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
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of KubeadmControlPlane.
`status`	`object`	status is the observed state of KubeadmControlPlane.

.spec

Description

spec is the desired state of KubeadmControlPlane.

Type

`object`

Required

`kubeadmConfigSpec` `machineTemplate` `version`

Property	Type	Description
`kubeadmConfigSpec`	`object`	kubeadmConfigSpec is a KubeadmConfigSpec to use for initializing and joining machines to the control plane.
`machineNamingStrategy`	`object`	machineNamingStrategy allows changing the naming pattern used when creating Machines. InfraMachines & KubeadmConfigs will use the same name as the corresponding Machines.
`machineTemplate`	`object`	machineTemplate contains information about how machines should be shaped when creating or updating a control plane.
`remediationStrategy`	`object`	remediationStrategy is the RemediationStrategy that controls how control plane machine remediation happens.
`replicas`	`integer`	replicas is the number of desired machines. Defaults to 1. When stacked etcd is used only odd numbers are permitted, as per etcd best practice ↗. This is a pointer to distinguish between explicit zero and not specified.
`rolloutAfter`	`string`	rolloutAfter is a field to indicate a rollout should be performed after the specified time even if no changes have been made to the KubeadmControlPlane. Example: In the YAML the time can be specified in the RFC3339 format. To specify the rolloutAfter target as March 9, 2023, at 9 am UTC use "2023-03-09T09:00:00Z".
`rolloutBefore`	`object`	rolloutBefore is a field to indicate a rollout should be performed if the specified criteria is met.
`rolloutStrategy`	`object`	rolloutStrategy is the RolloutStrategy to use to replace control plane machines with new ones.
`version`	`string`	version defines the desired Kubernetes version. Please note that if kubeadmConfigSpec.ClusterConfiguration.imageRepository is not set we don't allow upgrades to versions >= v1.22.0 for which kubeadm uses the old registry (k8s.gcr.io). Please use a newer patch version with the new registry instead. The default registries of kubeadm are: • registry.k8s.io (new registry): >= v1.22.17, >= v1.23.15, >= v1.24.9, >= v1.25.0 • k8s.gcr.io (old registry): all older versions

.spec.kubeadmConfigSpec

Description

kubeadmConfigSpec is a KubeadmConfigSpec to use for initializing and joining machines to the control plane.

Type

`object`

Property	Type	Description
`bootCommands`	`array`	bootCommands specifies extra commands to run very early in the boot process via the cloud-init bootcmd module. bootcmd will run on every boot, 'cloud-init-per' command can be used to make bootcmd run exactly once. This is typically run in the cloud-init.service systemd unit. This has no effect in Ignition.
`clusterConfiguration`	`object`	clusterConfiguration along with InitConfiguration are the configurations necessary for the init command
`diskSetup`	`object`	diskSetup specifies options for the creation of partition tables and file systems on devices.
`files`	`array`	files specifies extra files to be passed to user_data upon creation.
`format`	`string`	format specifies the output format of the bootstrap data
`ignition`	`object`	ignition contains Ignition specific configuration.
`initConfiguration`	`object`	initConfiguration along with ClusterConfiguration are the configurations necessary for the init command
`joinConfiguration`	`object`	joinConfiguration is the kubeadm configuration for the join command
`mounts`	`array`	mounts specifies a list of mount points to be setup.
`ntp`	`object`	ntp specifies NTP configuration
`postKubeadmCommands`	`array`	postKubeadmCommands specifies extra commands to run after kubeadm runs. With cloud-init, this is appended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is appended to /etc/kubeadm.sh.
`preKubeadmCommands`	`array`	preKubeadmCommands specifies extra commands to run before kubeadm runs. With cloud-init, this is prepended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is prepended to /etc/kubeadm.sh.
`useExperimentalRetryJoin`	`boolean`	useExperimentalRetryJoin replaces a basic kubeadm command with a shell script with retries for joins.

Property	Type	Description
		This is meant to be an experimental temporary workaround on some environments where joins fail due to timing (and other issues). The long term goal is to add retries to kubeadm proper and use that functionality. This will add about 40KB to userdata For more information, refer to https://github.com/kubernetes-sigs/cluster-api/pull/2763#discussion_r397306055 ↗. Deprecated: This experimental fix is no longer needed and this field will be removed in a future release. When removing also remove from staticcheck exclude-rules for SA1019 in golangci.yml
`users`	`array`	users specifies extra users to add
`verbosity`	`integer`	verbosity is the number for the kubeadm log level verbosity. It overrides the `--v` flag in kubeadm commands.

.spec.kubeadmConfigSpec.bootCommands

Description

bootCommands specifies extra commands to run very early in the boot process via the cloud-init bootcmd module. bootcmd will run on every boot, 'cloud-init-per' command can be used to make bootcmd run exactly once. This is typically run in the cloud-init.service systemd unit. This has no effect in Ignition.

Type

`array`

.spec.kubeadmConfigSpec.bootCommands[]

Type

`string`

.spec.kubeadmConfigSpec.clusterConfiguration

Description

clusterConfiguration along with InitConfiguration are the configurations necessary for the init command

Type

`object`

Property	Type	Description
`apiServer`	`object`	apiServer contains extra settings for the API server control plane component

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 ↗
`certificatesDir`	`string`	certificatesDir specifies where to store or look for all required certificates. NB: if not provided, this will default to `/etc/kubernetes/pki`
`clusterName`	`string`	clusterName is the cluster name
`controlPlaneEndpoint`	`string`	controlPlaneEndpoint sets a stable IP address or DNS name for the control plane; it can be a valid IP address or a RFC-1123 DNS subdomain, both with optional TCP port. In case the ControlPlaneEndpoint is not specified, the AdvertiseAddress + BindPort are used; in case the ControlPlaneEndpoint is specified but without a TCP port, the BindPort is used. Possible usages are: e.g. In a cluster with more than one control plane instances, this field should be assigned the address of the external load balancer in front of the control plane instances. e.g. in environments with enforced node recycling, the ControlPlaneEndpoint could be used for assigning a stable DNS to the control plane. NB: This value defaults to the first value in the Cluster object status.apiEndpoints array.
`controllerManager`	`object`	controllerManager contains extra settings for the controller manager control plane component
`dns`	`object`	dns defines the options for the DNS add-on installed in the cluster.
`etcd`	`object`	etcd holds configuration for etcd. NB: This value defaults to a Local (stacked) etcd
`featureGates`	`object`	featureGates enabled by the user.

`imageRepository`	`string`	imageRepository sets the container registry to pull images from.
		• If not set, the default registry of kubeadm will be used, i.e.
		  • registry.k8s.io (new registry): >= v1.22.17, >= v1.23.15, >= v1.24.9, >= v1.25.0
		  • k8s.gcr.io (old registry): all older versions Please note that when imageRepository is not set we don't allow upgrades to versions >= v1.22.0 which use the old registry (k8s.gcr.io). Please use a newer patch version with the new registry instead (i.e. >= v1.22.17,
		> = v1.23.15, >= v1.24.9, >= v1.25.0).
		• If the version is a CI build (kubernetes version starts with `ci/` or `ci-cross/`) `gcr.io/k8s-staging-ci-images` will be used as a default for control plane components and for kube-proxy, while `registry.k8s.io` will be used for all the other images.

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`kubernetesVersion`	`string`	kubernetesVersion is the target version of the control plane. NB: This value defaults to the Machine object spec.version
`networking`	`object`	networking holds configuration for the networking topology of the cluster. NB: This value defaults to the Cluster object spec.clusterNetwork.
`scheduler`	`object`	scheduler contains extra settings for the scheduler control plane component

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer

Description

apiServer contains extra settings for the API server control plane component

Type

`object`

Property	Type	Description
`certSANs`	`array`	certSANs sets extra Subject Alternative Names for the API Server signing cert.
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.
`timeoutForControlPlane`	`string`	timeoutForControlPlane controls the timeout that we use for API server to appear

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.certSANs

Description

certSANs sets extra Subject Alternative Names for the API Server signing cert.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.certSANs[]

Type

`string`

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[]

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.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].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.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.

Property	Type	Description
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager

Description

controllerManager contains extra settings for the controller manager control plane component

Type

`object`

Property	Type	Description
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[]

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.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].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.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.kubeadmConfigSpec.clusterConfiguration.dns

Description

dns defines the options for the DNS add-on installed in the cluster.

Type

`object`

Property	Type	Description
`imageRepository`	`string`	imageRepository sets the container registry to pull images from. if not set, the ImageRepository defined in ClusterConfiguration will be used instead.
`imageTag`	`string`	imageTag allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.

.spec.kubeadmConfigSpec.clusterConfiguration.etcd

Description

etcd holds configuration for etcd. NB: This value defaults to a Local (stacked) etcd

Type

`object`

Property	Type	Description
`external`	`object`	external describes how to connect to an external etcd cluster Local and External are mutually exclusive

Property	Type	Description
`local`	`object`	local provides configuration knobs for configuring the local etcd instance Local and External are mutually exclusive

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.external

Description

external describes how to connect to an external etcd cluster Local and External are mutually exclusive

Type

`object`

Required

`caFile` `certFile` `endpoints` `keyFile`

Property	Type	Description
`caFile`	`string`	caFile is an SSL Certificate Authority file used to secure etcd communication. Required if using a TLS connection.
`certFile`	`string`	certFile is an SSL certification file used to secure etcd communication. Required if using a TLS connection.
`endpoints`	`array`	endpoints of etcd members. Required for ExternalEtcd.
`keyFile`	`string`	keyFile is an SSL key file used to secure etcd communication. Required if using a TLS connection.

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.external.endpoints

Description

endpoints of etcd members. Required for ExternalEtcd.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.external.endpoints[]

Type

`string`

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local

Description

local provides configuration knobs for configuring the local etcd instance Local and External are mutually exclusive

Type

`object`

Property	Type	Description
`dataDir`	`string`	dataDir is the directory etcd will place its data. Defaults to "/var/lib/etcd".
`extraArgs`	`object`	extraArgs are extra arguments provided to the etcd binary when run inside a static pod.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`imageRepository`	`string`	imageRepository sets the container registry to pull images from. if not set, the ImageRepository defined in ClusterConfiguration will be used instead.
`imageTag`	`string`	imageTag allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.
`peerCertSANs`	`array`	peerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.
`serverCertSANs`	`array`	serverCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraArgs

Description

extraArgs are extra arguments provided to the etcd binary when run inside a static pod.

Type

`object`

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[]

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.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].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.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.peerCertSANs

Description

peerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.peerCertSANs[]

Type

`string`

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.serverCertSANs

Description

serverCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.serverCertSANs[]

Type

`string`

.spec.kubeadmConfigSpec.clusterConfiguration.featureGates

Description

featureGates enabled by the user.

Type

`object`

.spec.kubeadmConfigSpec.clusterConfiguration.networking

Description

networking holds configuration for the networking topology of the cluster. NB: This value defaults to the Cluster object spec.clusterNetwork.

Type

`object`

Property	Type	Description
`dnsDomain`	`string`	dnsDomain is the dns domain used by k8s services. Defaults to "cluster.local".
`podSubnet`	`string`	podSubnet is the subnet used by pods. If unset, the API server will not allocate CIDR ranges for every node. Defaults to a comma-delimited string of the Cluster object's spec.clusterNetwork.services.cidrBlocks if that is set
`serviceSubnet`	`string`	serviceSubnet is the subnet used by k8s services. Defaults to a comma-delimited string of the Cluster object's spec.clusterNetwork.pods.cidrBlocks, or to "10.96.0.0/12" if that's unset.

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler

Description

scheduler contains extra settings for the scheduler control plane component

Type

`object`

Property	Type	Description
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.

Property	Type	Description
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[]

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

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

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

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].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.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

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

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.kubeadmConfigSpec.diskSetup

Description

diskSetup specifies options for the creation of partition tables and file systems on devices.

Type

`object`

Property	Type	Description
`filesystems`	`array`	filesystems specifies the list of file systems to setup.
`partitions`	`array`	partitions specifies the list of the partitions to setup.

.spec.kubeadmConfigSpec.diskSetup.filesystems

Description

filesystems specifies the list of file systems to setup.

Type

`array`

.spec.kubeadmConfigSpec.diskSetup.filesystems[]

Description

Filesystem defines the file systems to be created.

Type

`object`

Required

`device` `filesystem`

Property	Type	Description
`device`	`string`	device specifies the device name
`extraOpts`	`array`	extraOpts defined extra options to add to the command for creating the file system.
`filesystem`	`string`	filesystem specifies the file system type.
`label`	`string`	label specifies the file system label to be used. If set to None, no label is used.
`overwrite`	`boolean`	overwrite defines whether or not to overwrite any existing filesystem. If true, any pre-existing file system will be destroyed. Use with Caution.
`partition`	`string`	partition specifies the partition to use. The valid options are: "auto|any", "auto", "any", "none", and , where NUM is the actual partition number.
`replaceFS`	`string`	replaceFS is a special directive, used for Microsoft Azure that instructs cloud-init to replace a file system of <FS_TYPE>. NOTE: unless you define a label, this requires the use of the 'any' partition directive.

.spec.kubeadmConfigSpec.diskSetup.filesystems[].extraOpts

Description

extraOpts defined extra options to add to the command for creating the file system.

Type

`array`

.spec.kubeadmConfigSpec.diskSetup.filesystems[].extraOpts[]

Type

`string`

.spec.kubeadmConfigSpec.diskSetup.partitions

Description

partitions specifies the list of the partitions to setup.

Type

`array`

.spec.kubeadmConfigSpec.diskSetup.partitions[]

Description

Partition defines how to create and layout a partition.

Type

`object`

Required

`device` `layout`

Property	Type	Description
`device`	`string`	device is the name of the device.
`layout`	`boolean`	layout specifies the device layout. If it is true, a single partition will be created for the entire device. When layout is false, it means don't partition or ignore existing partitioning.
`overwrite`	`boolean`	overwrite describes whether to skip checks and create the partition if a partition or filesystem is found on the device. Use with caution. Default is 'false'.
`tableType`	`string`	tableType specifies the tupe of partition table. The following are supported: 'mbr': default and setups a MS-DOS partition table 'gpt': setups a GPT partition table

.spec.kubeadmConfigSpec.files

Description

files specifies extra files to be passed to user_data upon creation.

Type

`array`

.spec.kubeadmConfigSpec.files[]

Description

File defines the input for generating write_files in cloud-init.

Type

`object`

Required

`path`

Property	Type	Description
`append`	`boolean`	append specifies whether to append Content to existing file if Path exists.
`content`	`string`	content is the actual content of the file.
`contentFrom`	`object`	contentFrom is a referenced source of content to populate the file.

Property	Type	Description
`encoding`	`string`	encoding specifies the encoding of the file contents.
`owner`	`string`	owner specifies the ownership of the file, e.g. "root:root".
`path`	`string`	path specifies the full path on disk where to store the file.
`permissions`	`string`	permissions specifies the permissions to assign to the file, e.g. "0640".

.spec.kubeadmConfigSpec.files[].contentFrom

Description

contentFrom is a referenced source of content to populate the file.

Type

`object`

Required

`secret`

Property	Type	Description
`secret`	`object`	secret represents a secret that should populate this file.

.spec.kubeadmConfigSpec.files[].contentFrom.secret

Description

secret represents a secret that should populate this file.

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	key is the key in the secret's data map for this value.
`name`	`string`	name of the secret in the KubeadmBootstrapConfig's namespace to use.

.spec.kubeadmConfigSpec.ignition

Description

ignition contains Ignition specific configuration.

Type

`object`

Property	Type	Description
`containerLinuxConfig`	`object`	containerLinuxConfig contains CLC specific configuration.

.spec.kubeadmConfigSpec.ignition.containerLinuxConfig

Description

containerLinuxConfig contains CLC specific configuration.

Type

`object`

Property	Type	Description
`additionalConfig`	`string`	additionalConfig contains additional configuration to be merged with the Ignition configuration generated by the bootstrapper controller. More info: https://coreos.github.io/ignition/operator-notes/#config-merging ↗ The data format is documented here: https://kinvolk.io/docs/flatcar-container-linux/latest/provisioning/cl-config/ ↗
`strict`	`boolean`	strict controls if AdditionalConfig should be strictly parsed. If so, warnings are treated as errors.

.spec.kubeadmConfigSpec.initConfiguration

Description

initConfiguration along with ClusterConfiguration are the configurations necessary for the init command

Type

`object`

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 ↗
`bootstrapTokens`	`array`	bootstrapTokens is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`localAPIEndpoint`	`object`	localAPIEndpoint represents the endpoint of the API server instance that's deployed on this control plane node In HA setups, this differs from ClusterConfiguration.ControlPlaneEndpoint in the sense that ControlPlaneEndpoint is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.
`nodeRegistration`	`object`	nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration
`patches`	`object`	patches contains options related to applying patches to components deployed by kubeadm during "kubeadm init". The minimum kubernetes version needed to support Patches is v1.22
`skipPhases`	`array`	skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens

Description

bootstrapTokens is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature

Type

`array`

.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[]

Description

BootstrapToken describes one bootstrap token, stored as a Secret in the cluster.

Type

`object`

Required

`token`

Property	Type	Description
`description`	`string`	description sets a human-friendly message why this token exists and what it's used for, so other administrators can know its purpose.
`expires`	`string`	expires specifies the timestamp when this token expires. Defaults to being set dynamically at runtime based on the TTL. Expires and TTL are mutually exclusive.
`groups`	`array`	groups specifies the extra groups that this token will authenticate as when/if used for authentication
`token`	`string`	token is used for establishing bidirectional trust between nodes and control-planes. Used for joining nodes in the cluster.
`ttl`	`string`	ttl defines the time to live for this token. Defaults to 24h. Expires and TTL are mutually exclusive.
`usages`	`array`	usages describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.

.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[].groups

Description

groups specifies the extra groups that this token will authenticate as when/if used for authentication

Type

`array`

.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[].groups[]

Type

`string`

.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[].usages

Description

usages describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.

Type

`array`

.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[].usages[]

Type

`string`

.spec.kubeadmConfigSpec.initConfiguration.localAPIEndpoint

Description

localAPIEndpoint represents the endpoint of the API server instance that's deployed on this control plane node In HA setups, this differs from ClusterConfiguration.ControlPlaneEndpoint in the sense that ControlPlaneEndpoint is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.

Type

`object`

Property	Type	Description
`advertiseAddress`	`string`	advertiseAddress sets the IP address for the API server to advertise.
`bindPort`	`integer`	bindPort sets the secure port for the API Server to bind to. Defaults to 6443.

.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration

Description

nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Type

`object`

Property	Type	Description
`criSocket`	`string`	criSocket is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use
`ignorePreflightErrors`	`array`	ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.
`imagePullPolicy`	`string`	imagePullPolicy specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". Defaults to "IfNotPresent". This can be used only with Kubernetes version equal to 1.22 and later.
`imagePullSerial`	`boolean`	imagePullSerial specifies if image pulling performed by kubeadm must be done serially or in parallel. This option takes effect only on Kubernetes >=1.31.0. Default: true (defaulted in kubeadm)

Property	Type	Description
`kubeletExtraArgs`	`object`	kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.
`name`	`string`	name is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the CommonName field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`taints`	`array`	taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.ignorePreflightErrors

Description

ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Type

`array`

.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.ignorePreflightErrors[]

Type

`string`

.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.kubeletExtraArgs

Description

kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.

Type

`object`

.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.taints

Description

taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

Type

`array`

.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.kubeadmConfigSpec.initConfiguration.patches

Description

patches contains options related to applying patches to components deployed by kubeadm during "kubeadm init". The minimum kubernetes version needed to support Patches is v1.22

Type

`object`

Property	Type	Description
`directory`	`string`	directory is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic" "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically. These files can be written into the target directory via KubeadmConfig.Files which specifies additional files to be created on the machine, either with content inline or by referencing a secret.

.spec.kubeadmConfigSpec.initConfiguration.skipPhases

Description

skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

Type

`array`

.spec.kubeadmConfigSpec.initConfiguration.skipPhases[]

Type

`string`

.spec.kubeadmConfigSpec.joinConfiguration

Description

joinConfiguration is the kubeadm configuration for the join command

Type

`object`

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 ↗
`caCertPath`	`string`	caCertPath is the path to the SSL certificate authority used to secure comunications between node and control-plane. Defaults to "/etc/kubernetes/pki/ca.crt".
`controlPlane`	`object`	controlPlane defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.
`discovery`	`object`	discovery specifies the options for the kubelet to use during the TLS Bootstrap process
`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 ↗
`nodeRegistration`	`object`	nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration
`patches`	`object`	patches contains options related to applying patches to components deployed by kubeadm during "kubeadm join". The minimum kubernetes version needed to support Patches is v1.22
`skipPhases`	`array`	skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

.spec.kubeadmConfigSpec.joinConfiguration.controlPlane

Description

controlPlane defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.

Type

`object`

Property	Type	Description
`localAPIEndpoint`	`object`	localAPIEndpoint represents the endpoint of the API server instance to be deployed on this node.

.spec.kubeadmConfigSpec.joinConfiguration.controlPlane.localAPIEndpoint

Description

localAPIEndpoint represents the endpoint of the API server instance to be deployed on this node.

Type

`object`

Property	Type	Description
`advertiseAddress`	`string`	advertiseAddress sets the IP address for the API server to advertise.
`bindPort`	`integer`	bindPort sets the secure port for the API Server to bind to. Defaults to 6443.

.spec.kubeadmConfigSpec.joinConfiguration.discovery

Description

discovery specifies the options for the kubelet to use during the TLS Bootstrap process

Type

`object`

Property	Type	Description
`bootstrapToken`	`object`	bootstrapToken is used to set the options for bootstrap token based discovery BootstrapToken and File are mutually exclusive
`file`	`object`	file is used to specify a file or URL to a kubeconfig file from which to load cluster information BootstrapToken and File are mutually exclusive
`timeout`	`string`	timeout modifies the discovery timeout

Property	Type	Description
`tlsBootstrapToken`	`string`	tlsBootstrapToken is a token used for TLS bootstrapping. If .BootstrapToken is set, this field is defaulted to .BootstrapToken.Token, but can be overridden. If .File is set, this field **must be set** in case the KubeConfigFile does not contain any other authentication information

.spec.kubeadmConfigSpec.joinConfiguration.discovery.bootstrapToken

Description

bootstrapToken is used to set the options for bootstrap token based discovery BootstrapToken and File are mutually exclusive

Type

`object`

Property	Type	Description
`apiServerEndpoint`	`string`	apiServerEndpoint is an IP or domain name to the API server from which info will be fetched.
`caCertHashes`	`array`	caCertHashes specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as ":", where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL: openssl x509 -pubkey -in ca.crt openssl rsa -pubin -outform der 2>&/dev/null | openssl dgst -sha256 -hex
`token`	`string`	token is a token used to validate cluster information fetched from the control-plane.
`unsafeSkipCAVerification`	`boolean`	unsafeSkipCAVerification allows token-based discovery without CA verification via CACertHashes. This can weaken the security of kubeadm since other nodes can impersonate the control-plane.

.spec.kubeadmConfigSpec.joinConfiguration.discovery.bootstrapToken.caCertHashes

Description

caCertHashes specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as "<type>:<value>", where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL: openssl x509 -pubkey -in ca.crt openssl rsa -pubin -outform der 2>&/dev/null | openssl dgst -sha256 -hex

Type

`array`

.spec.kubeadmConfigSpec.joinConfiguration.discovery.bootstrapToken.caCertHashes[]

Type

`string`

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file

Description

file is used to specify a file or URL to a kubeconfig file from which to load cluster information BootstrapToken and File are mutually exclusive

Type

`object`

Required

`kubeConfigPath`

Property	Type	Description
`kubeConfig`	`object`	kubeConfig is used (optionally) to generate a KubeConfig based on the KubeadmConfig's information. The file is generated at the path specified in KubeConfigPath. Host address (server field) information is automatically populated based on the Cluster's ControlPlaneEndpoint. Certificate Authority (certificate-authority-data field) is gathered from the cluster's CA secret.
`kubeConfigPath`	`string`	kubeConfigPath is used to specify the actual file path or URL to the kubeconfig file from which to load cluster information

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig

Description

kubeConfig is used (optionally) to generate a KubeConfig based on the KubeadmConfig's information. The file is generated at the path specified in KubeConfigPath. Host address (server field) information is automatically populated based on the Cluster's ControlPlaneEndpoint. Certificate Authority (certificate-authority-data field) is gathered from the cluster's CA secret.

Type

`object`

Required

`user`

Property	Type	Description
`cluster`	`object`	cluster contains information about how to communicate with the kubernetes cluster. By default the following fields are automatically populated: • Server with the Cluster's ControlPlaneEndpoint. • CertificateAuthorityData with the Cluster's CA certificate.
`user`	`object`	user contains information that describes identity information. This is used to tell the kubernetes cluster who you are.

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.cluster

Description

cluster contains information about how to communicate with the kubernetes cluster. By default the following fields are automatically populated: - Server with the Cluster's ControlPlaneEndpoint. - CertificateAuthorityData with the Cluster's CA certificate.

Type

`object`

Property	Type	Description
`certificateAuthorityData`	`string`	certificateAuthorityData contains PEM-encoded certificate authority certificates. Defaults to the Cluster's CA certificate if empty.
`insecureSkipTLSVerify`	`boolean`	insecureSkipTLSVerify skips the validity check for the server's certificate. This will make your HTTPS connections insecure.
`proxyURL`	`string`	proxyURL is the URL to the proxy to be used for all requests made by this client. URLs with "http", "https", and "socks5" schemes are supported. If this configuration is not provided or the empty string, the client attempts to construct a proxy configuration from http_proxy and https_proxy environment variables. If these environment variables are not set, the client does not attempt to proxy requests. socks5 proxying does not currently support spdy streaming endpoints (exec, attach, port forward).
`server`	`string`	server is the address of the kubernetes cluster (https://hostname:port ↗). Defaults to https:// + Cluster.Spec.ControlPlaneEndpoint.
`tlsServerName`	`string`	tlsServerName is used to check server certificate. If TLSServerName is empty, the hostname used to contact the server is used.

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user

Description

user contains information that describes identity information. This is used to tell the kubernetes cluster who you are.

Type

`object`

Property	Type	Description
`authProvider`	`object`	authProvider specifies a custom authentication plugin for the kubernetes cluster.
`exec`	`object`	exec specifies a custom exec-based authentication plugin for the kubernetes cluster.

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.authProvider

Description

authProvider specifies a custom authentication plugin for the kubernetes cluster.

Type

`object`

Required

`name`

Property	Type	Description
`config`	`object`	config holds the parameters for the authentication plugin.
`name`	`string`	name is the name of the authentication plugin.

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.authProvider.config

Description

config holds the parameters for the authentication plugin.

Type

`object`

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec

Description

exec specifies a custom exec-based authentication plugin for the kubernetes cluster.

Type

`object`

Required

`command`

Property	Type	Description
`apiVersion`	`string`	apiVersion is preferred input version of the ExecInfo. The returned ExecCredentials MUST use the same encoding version as the input. Defaults to client.authentication.k8s.io/v1 if not set.
`args`	`array`	args is the arguments to pass to the command when executing it.
`command`	`string`	command to execute.
`env`	`array`	env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

Property	Type	Description
`provideClusterInfo`	`boolean`	provideClusterInfo determines whether or not to provide cluster information, which could potentially contain very large CA data, to this exec plugin as a part of the KUBERNETES_EXEC_INFO environment variable. By default, it is set to false. Package k8s.io/client-go/tools/auth/exec provides helper methods for reading this environment variable.

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec.args

Description

args is the arguments to pass to the command when executing it.

Type

`array`

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec.args[]

Type

`string`

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec.env

Description

env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

Type

`array`

.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec.env[]

Description

KubeConfigAuthExecEnv is used for setting environment variables when executing an exec-based credential plugin.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	name of the environment variable
`value`	`string`	value of the environment variable

.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration

Description

nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Type

`object`

Property	Type	Description
`criSocket`	`string`	criSocket is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use
`ignorePreflightErrors`	`array`	ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.
`imagePullPolicy`	`string`	imagePullPolicy specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". Defaults to "IfNotPresent". This can be used only with Kubernetes version equal to 1.22 and later.
`imagePullSerial`	`boolean`	imagePullSerial specifies if image pulling performed by kubeadm must be done serially or in parallel. This option takes effect only on Kubernetes >=1.31.0. Default: true (defaulted in kubeadm)
`kubeletExtraArgs`	`object`	kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.
`name`	`string`	name is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the CommonName field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`taints`	`array`	taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.ignorePreflightErrors

Description

ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Type

`array`

.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.ignorePreflightErrors[]

Type

`string`

.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.kubeletExtraArgs

Description

kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.

Type

`object`

.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.taints

Description

taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""'}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

Type

`array`

.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.kubeadmConfigSpec.joinConfiguration.patches

Description

patches contains options related to applying patches to components deployed by kubeadm during "kubeadm join". The minimum kubernetes version needed to support Patches is v1.22

Type

`object`

Property	Type	Description
`directory`	`string`	directory is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic" "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically. These files can be written into the target directory via KubeadmConfig.Files which specifies additional files to be created on the machine, either with content inline or by referencing a secret.

.spec.kubeadmConfigSpec.joinConfiguration.skipPhases

Description

skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

Type

`array`

.spec.kubeadmConfigSpec.joinConfiguration.skipPhases[]

Type

`string`

.spec.kubeadmConfigSpec.mounts

Description

mounts specifies a list of mount points to be setup.

Type

`array`

.spec.kubeadmConfigSpec.mounts[]

Description

MountPoints defines input for generated mounts in cloud-init.

Type

`array`

.spec.kubeadmConfigSpec.ntp

Description

ntp specifies NTP configuration

Type

`object`

Property	Type	Description
`enabled`	`boolean`	enabled specifies whether NTP should be enabled
`servers`	`array`	servers specifies which NTP servers to use

.spec.kubeadmConfigSpec.ntp.servers

Description

servers specifies which NTP servers to use

Type

`array`

.spec.kubeadmConfigSpec.ntp.servers[]

Type

`string`

.spec.kubeadmConfigSpec.postKubeadmCommands

Description

postKubeadmCommands specifies extra commands to run after kubeadm runs. With cloud-init, this is appended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is appended to /etc/kubeadm.sh.

Type

`array`

.spec.kubeadmConfigSpec.postKubeadmCommands[]

Type

`string`

.spec.kubeadmConfigSpec.preKubeadmCommands

Description

preKubeadmCommands specifies extra commands to run before kubeadm runs. With cloud-init, this is prepended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is prepended to /etc/kubeadm.sh.

Type

`array`

.spec.kubeadmConfigSpec.preKubeadmCommands[]

Type

`string`

.spec.kubeadmConfigSpec.users

Description

users specifies extra users to add

Type

`array`

.spec.kubeadmConfigSpec.users[]

Description

User defines the input for a generated user in cloud-init.

Type

`object`

Required

`name`

Property	Type	Description
`gecos`	`string`	gecos specifies the gecos to use for the user
`groups`	`string`	groups specifies the additional groups for the user
`homeDir`	`string`	homeDir specifies the home directory to use for the user
`inactive`	`boolean`	inactive specifies whether to mark the user as inactive
`lockPassword`	`boolean`	lockPassword specifies if password login should be disabled
`name`	`string`	name specifies the user name
`passwd`	`string`	passwd specifies a hashed password for the user
`passwdFrom`	`object`	passwdFrom is a referenced source of passwd to populate the passwd.
`primaryGroup`	`string`	primaryGroup specifies the primary group for the user
`shell`	`string`	shell specifies the user's shell
`sshAuthorizedKeys`	`array`	sshAuthorizedKeys specifies a list of ssh authorized keys for the user

Property	Type	Description
`sudo`	`string`	sudo specifies a sudo role for the user

.spec.kubeadmConfigSpec.users[].passwdFrom

Description

passwdFrom is a referenced source of passwd to populate the passwd.

Type

`object`

Required

`secret`

Property	Type	Description
`secret`	`object`	secret represents a secret that should populate this password.

.spec.kubeadmConfigSpec.users[].passwdFrom.secret

Description

secret represents a secret that should populate this password.

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	key is the key in the secret's data map for this value.
`name`	`string`	name of the secret in the KubeadmBootstrapConfig's namespace to use.

.spec.kubeadmConfigSpec.users[].sshAuthorizedKeys

Description

sshAuthorizedKeys specifies a list of ssh authorized keys for the user

Type

`array`

.spec.kubeadmConfigSpec.users[].sshAuthorizedKeys[]

Type

`string`

.spec.machineNamingStrategy

Description

machineNamingStrategy allows changing the naming pattern used when creating Machines. InfraMachines & KubeadmConfigs will use the same name as the corresponding Machines.

Type

`object`

Property	Type	Description
`template`	`string`	template defines the template to use for generating the names of the Machine objects. If not defined, it will fallback to `{{ .kubeadmControlPlane.name }}-{{ .random }}`. If the generated name string exceeds 63 characters, it will be trimmed to 58 characters and will get concatenated with a random suffix of length 5. Length of the template string must not exceed 256 characters. The template allows the following variables `.cluster.name`, `.kubeadmControlPlane.name` and `.random`. The variable `.cluster.name` retrieves the name of the cluster object that owns the Machines being created. The variable `.kubeadmControlPlane.name` retrieves the name of the KubeadmControlPlane object that owns the Machines being created. The variable `.random` is substituted with random alphanumeric string, without vowels, of length 5. This variable is required part of the template. If not provided, validation will fail.

.spec.machineTemplate

Description

machineTemplate contains information about how machines should be shaped when creating or updating a control plane.

Type

`object`

Required

`infrastructureRef`

Property	Type	Description
`infrastructureRef`	`object`	infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.
`metadata`	`ObjectMeta`	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the machine controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. If no value is provided, the default value for this property of the Machine resource will be used.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a controlplane node The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`

Property	Type	Description
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.
`readinessGates`	`array`	readinessGates specifies additional conditions to include when evaluating Machine Ready condition; KubeadmControlPlane will always add readinessGates for the condition it is setting on the Machine: APIServerPodHealthy, SchedulerPodHealthy, ControllerManagerPodHealthy, and if etcd is managed by CKP also EtcdPodHealthy, EtcdMemberHealthy. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. NOTE: This field is considered only for computing v1beta2 conditions.
`taints`	`array`	The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints

.spec.machineTemplate.infrastructureRef

Description

infrastructureRef is a required reference to a custom resource offered by an infrastructure provider.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.machineTemplate.readinessGates

Description

readinessGates specifies additional conditions to include when evaluating Machine Ready condition; KubeadmControlPlane will always add readinessGates for the condition it is setting on the Machine: APIServerPodHealthy, SchedulerPodHealthy, ControllerManagerPodHealthy, and if etcd is managed by CKP also EtcdPodHealthy, EtcdMemberHealthy. This field can be used e.g. to instruct the machine controller to include in the computation for Machine's ready computation a condition, managed by an external controllers, reporting the status of special software/hardware installed on the Machine. NOTE: This field is considered only for computing v1beta2 conditions.

Type

`array`

.spec.machineTemplate.readinessGates[]

Description

MachineReadinessGate contains the type of a Machine condition to be used as a readiness gate.

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	conditionType refers to a condition with matching type in the Machine's condition list. If the conditions doesn't exist, it will be treated as unknown. Note: Both Cluster API conditions or conditions added by 3rd party controllers can be used as readiness gates.

Property	Type	Description
`polarity`	`string`	polarity of the conditionType specified in this readinessGate. Valid values are Positive, Negative and omitted. When omitted, the default behaviour will be Positive. A positive polarity means that the condition should report a true status under normal conditions. A negative polarity means that the condition should report a false status under normal conditions.

.spec.machineTemplate.taints

Description

The list of the taints to be applied to the corresponding Node in additive manner. This list will not overwrite any other taints added to the Node on an ongoing basis by other entities. These taints should be actively reconciled e.g. if you ask the machine controller to apply a taint and then manually remove the taint the machine controller will put it back) but not have the machine controller remove any taints

Type

`array`

.spec.machineTemplate.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.remediationStrategy

Description

remediationStrategy is the RemediationStrategy that controls how control plane machine remediation happens.

Type

`object`

Property	Type	Description

`maxRetry`	`integer`	maxRetry is the Max number of retries while attempting to remediate an unhealthy machine. A retry happens when a created as a replacement for an unhealthy machine also fails. For example, given a control plane with three machine
		`M1 become unhealthy; remediation happens, and M1-1 is created as a replacement.`
		`If M1-1 (replacement of M1) has problems while bootstrapping it will become unhealthy`
		`remediated; such operation is considered a retry, remediation-retry #1.`
		`If M1-2 (replacement of M1-1) becomes unhealthy, remediation-retry #2 will happen, et`
		A retry could happen only after RetryPeriod from the previous retry. If a machine is marked as unhealthy after MinHe the previous remediation expired, this is not considered a retry anymore because the new issue is assumed unrelate one.
		If not set, the remedation will be retried infinitely.

`minHealthyPeriod`	`string`	minHealthyPeriod defines the duration after which KCP will consider any failure to a machine unrelated from the pre case the remediation is not considered a retry anymore, and thus the retry counter restarts from 0. For example, ass MinHealthyPeriod is set to 1h (default)
		`M1 become unhealthy; remediation happens, and M1-1 is created as a replacement.`
		`If M1-1 (replacement of M1) has problems within the 1hr after the creation, also`
		`this machine will be remediated and this operation is considered a retry - a problem`
		`to the original issue happened to M1 -.`
		`If instead the problem on M1-1 is happening after MinHealthyPeriod expired, e.g. four`
		`m1-1 has been created as a remediation of M1, the problem on M1-1 is considered unrel`
		`the original issue happened to M1.`
		If not set, this value is defaulted to 1h.

`retryPeriod`	`string`	retryPeriod is the duration that KCP should wait before remediating a machine being created as a replacement for a machine (a retry). If not set, a retry will happen immediately.

.spec.rolloutBefore

Description

rolloutBefore is a field to indicate a rollout should be performed if the specified criteria is met.

Type

`object`

Property	Type	Description
`certificatesExpiryDays`	`integer`	certificatesExpiryDays indicates a rollout needs to be performed if the certificates of the machine will expire within the specified days.

.spec.rolloutStrategy

Description

rolloutStrategy is the RolloutStrategy to use to replace control plane machines with new ones.

Type

`object`

Property	Type	Description
`rollingUpdate`	`object`	rollingUpdate is the rolling update config params. Present only if RolloutStrategyType = RollingUpdate.
`type`	`string`	type of rollout. Currently the only supported strategy is "RollingUpdate". Default is RollingUpdate.

.spec.rolloutStrategy.rollingUpdate

Description

rollingUpdate is the rolling update config params. Present only if RolloutStrategyType = RollingUpdate.

Type

`object`

Property	Type	Description
`maxSurge`	`integer|string`	maxSurge is the maximum number of control planes that can be scheduled above or under the desired number of control planes. Value can be an absolute number 1 or 0. Defaults to 1. Example: when this is set to 1, the control plane can be scaled up immediately when the rolling update starts.

.status

Description

status is the observed state of KubeadmControlPlane.

Type

`object`

Property	Type	Description
`conditions`	`array`	conditions defines current service state of the KubeadmControlPlane.
`failureMessage`	`string`	failureMessage indicates that there is a terminal problem reconciling the state, and will be set to a descriptive error message. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.

Property	Type	Description
`failureReason`	`string`	failureReason indicates that there is a terminal problem reconciling the state, and will be set to a token value suitable for programmatic interpretation. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.
`initialized`	`boolean`	initialized denotes that the KubeadmControlPlane API Server is initialized and thus it can accept requests. NOTE: this field is part of the Cluster API contract and it is used to orchestrate provisioning. The value of this field is never updated after provisioning is completed. Please use conditions to check the operational state of the control plane.
`lastRemediation`	`object`	lastRemediation stores info about last remediation performed.
`observedGeneration`	`integer`	observedGeneration is the latest generation observed by the controller.
`ready`	`boolean`	ready denotes that the KubeadmControlPlane API Server became ready during initial provisioning to receive requests. NOTE: this field is part of the Cluster API contract and it is used to orchestrate provisioning. The value of this field is never updated after provisioning is completed. Please use conditions to check the operational state of the control plane.
`readyReplicas`	`integer`	readyReplicas is the total number of fully running and ready control plane machines.
`replicas`	`integer`	replicas is the total number of non-terminated machines targeted by this control plane (their labels match the selector).
`selector`	`string`	selector is the label selector in string format to avoid introspection by clients, and is used to provide the CRD-based integration for the scale subresource and additional integrations for things like kubectl describe.. The string will be in the same format as the query-param syntax. More info about label selectors: http://kubernetes.io/docs/user-guide/labels#label-selectors ↗
`unavailableReplicas`	`integer`	unavailableReplicas is the total number of unavailable machines targeted by this control plane. This is the total number of machines that are still required for the deployment to have 100% available capacity. They may either be machines that are running but not yet ready or machines that still have not been created. Deprecated: This field is deprecated and is going to be removed in the next apiVersion. Please see https://github.com/kubernetes-sigs/cluster-api/blob/main/docs/proposals/20240916-improve-status-in-CAPI-resources.md ↗ for more details.

Property	Type	Description
`updatedReplicas`	`integer`	updatedReplicas is the total number of non-terminated machines targeted by this control plane that have the desired template spec.
`v1beta2`	`object`	v1beta2 groups all the fields that will be added or modified in KubeadmControlPlane's status with the V1Beta2 version.
`version`	`string`	version represents the minimum Kubernetes version for the control plane machines in the cluster.

.status.conditions

Description

conditions defines current service state of the KubeadmControlPlane.

Type

`array`

.status.conditions[]

Description

Condition defines an observation of a Cluster API resource operational state.

Type

`object`

Required

`lastTransitionTime` `status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This field may be empty.
`reason`	`string`	reason is the reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may be empty.
`severity`	`string`	severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.

Property	Type	Description
`status`	`string`	status of the condition, one of True, False, Unknown.
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase. Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important.

.status.lastRemediation

Description

lastRemediation stores info about last remediation performed.

Type

`object`

Required

`machine` `retryCount` `timestamp`

Property	Type	Description
`machine`	`string`	machine is the machine name of the latest machine being remediated.
`retryCount`	`integer`	retryCount used to keep track of remediation retry for the last remediated machine. A retry happens when a machine that was created as a replacement for an unhealthy machine also fails.
`timestamp`	`string`	timestamp is when last remediation happened. It is represented in RFC3339 form and is in UTC.

.status.v1beta2

Description

v1beta2 groups all the fields that will be added or modified in KubeadmControlPlane's status with the V1Beta2 version.

Type

`object`

Property	Type	Description
`availableReplicas`	`integer`	availableReplicas is the number of available replicas targeted by this KubeadmControlPlane. A machine is considered available when Machine's Available condition is true.
`conditions`	`array`	conditions represents the observations of a KubeadmControlPlane's current state. Known condition types are Available, CertificatesAvailable, EtcdClusterAvailable, MachinesReady, MachinesUpToDate, ScalingUp, ScalingDown, Remediating, Deleting, Paused.

Property	Type	Description
`readyReplicas`	`integer`	readyReplicas is the number of ready replicas for this KubeadmControlPlane. A machine is considered ready when Machine's Ready condition is true.
`upToDateReplicas`	`integer`	upToDateReplicas is the number of up-to-date replicas targeted by this KubeadmControlPlane. A machine is considered up-to-date when Machine's UpToDate condition is true.

.status.v1beta2.conditions

Description

conditions represents the observations of a KubeadmControlPlane's current state. Known condition types are Available, CertificatesAvailable, EtcdClusterAvailable, MachinesReady, MachinesUpToDate, ScalingUp, ScalingDown, Remediating, Deleting, Paused.

Type

`array`

.status.v1beta2.conditions[]

Description

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

Type

`object`

Required

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

Property	Type	Description
`lastTransitionTime`	`string`	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
`message`	`string`	message is a human readable message indicating details about the transition. This may be an empty string.
`observedGeneration`	`integer`	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
`reason`	`string`	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
`status`	`string`	status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	type of condition in CamelCase or in foo.example.com/CamelCase.

API Endpoints

The following API endpoints are available:

- `/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanes`
  - `DELETE` : delete collection of KubeadmControlPlane
  - `GET` : list objects of kind KubeadmControlPlane
  - `POST` : create a new KubeadmControlPlane
- `/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanes/{name}`
  - `DELETE` : delete the specified KubeadmControlPlane
  - `GET` : read the specified KubeadmControlPlane
  - `PATCH` : partially update the specified KubeadmControlPlane
  - `PUT` : replace the specified KubeadmControlPlane
- `/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanes/{name}/status`
  - `GET` : read status of the specified KubeadmControlPlane
  - `PATCH` : partially update status of the specified KubeadmControlPlane
  - `PUT` : replace status of the specified KubeadmControlPlane
- `/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanes/{name}/scale`
  - `GET` : read scale of the specified KubeadmControlPlane
  - `PATCH` : partially update scale of the specified KubeadmControlPlane
  - `PUT` : replace scale of the specified KubeadmControlPlane

/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanes

HTTP method

`DELETE`

Description

delete collection of KubeadmControlPlane

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind KubeadmControlPlane

HTTP responses

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

HTTP method

`POST`

Description

create a new KubeadmControlPlane

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`KubeadmControlPlane` schema	`application/json` formatted

HTTP responses

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

/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanes/{name}

HTTP method

`DELETE`

Description

delete the specified KubeadmControlPlane

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 KubeadmControlPlane

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified KubeadmControlPlane

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified KubeadmControlPlane

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`KubeadmControlPlane` schema	`application/json` formatted

HTTP responses

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

/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanes/{name}/status

HTTP method

`GET`

Description

read status of the specified KubeadmControlPlane

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified KubeadmControlPlane

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified KubeadmControlPlane

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`KubeadmControlPlane` schema	`application/json` formatted

HTTP responses

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

/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanes/{name}/scale

HTTP method

`GET`

Description

read scale of the specified KubeadmControlPlane

HTTP responses

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

HTTP method

`PATCH`

Description

partially update scale of the specified KubeadmControlPlane

Query parameters

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

HTTP method

`PUT`

Description

replace scale of the specified KubeadmControlPlane

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`Scale` schema	`application/json` formatted

KubeadmControlPlaneTemplate [controlplane.cluster.x-k8s.io/v1beta1]

Description

KubeadmControlPlaneTemplate is the Schema for the kubeadmcontrolplanetemplates 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
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	spec is the desired state of KubeadmControlPlaneTemplate.

.spec

Description

spec is the desired state of KubeadmControlPlaneTemplate.

Type

`object`

Required

`template`

Property	Type	Description
`template`	`object`	template defines the desired state of KubeadmControlPlaneTemplate.

.spec.template

Description

template defines the desired state of KubeadmControlPlaneTemplate.

Type

`object`

Required

`spec`

Property	Type	Description
`metadata`	`ObjectMeta`	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`spec`	`object`	spec is the desired state of KubeadmControlPlaneTemplateResource.

.spec.template.spec

Description

spec is the desired state of KubeadmControlPlaneTemplateResource.

Type

`object`

Required

`kubeadmConfigSpec`

Property	Type	Description
`kubeadmConfigSpec`	`object`	kubeadmConfigSpec is a KubeadmConfigSpec to use for initializing and joining machines to the control plane.
`machineNamingStrategy`	`object`	machineNamingStrategy allows changing the naming pattern used when creating Machines. InfraMachines & KubeadmConfigs will use the same name as the corresponding Machines.
`machineTemplate`	`object`	machineTemplate contains information about how machines should be shaped when creating or updating a control plane.

Property	Type	Description
`remediationStrategy`	`object`	remediationStrategy is the RemediationStrategy that controls how control plane machine remediation happens.
`rolloutAfter`	`string`	rolloutAfter is a field to indicate a rollout should be performed after the specified time even if no changes have been made to the KubeadmControlPlane.
`rolloutBefore`	`object`	rolloutBefore is a field to indicate a rollout should be performed if the specified criteria is met.
`rolloutStrategy`	`object`	rolloutStrategy is the RolloutStrategy to use to replace control plane machines with new ones.

.spec.template.spec.kubeadmConfigSpec

Description

kubeadmConfigSpec is a KubeadmConfigSpec to use for initializing and joining machines to the control plane.

Type

`object`

Property	Type	Description
`bootCommands`	`array`	bootCommands specifies extra commands to run very early in the boot process via the cloud-init bootcmd module. bootcmd will run on every boot, 'cloud-init-per' command can be used to make bootcmd run exactly once. This is typically run in the cloud-init.service systemd unit. This has no effect in Ignition.
`clusterConfiguration`	`object`	clusterConfiguration along with InitConfiguration are the configurations necessary for the init command
`diskSetup`	`object`	diskSetup specifies options for the creation of partition tables and file systems on devices.
`files`	`array`	files specifies extra files to be passed to user_data upon creation.
`format`	`string`	format specifies the output format of the bootstrap data
`ignition`	`object`	ignition contains Ignition specific configuration.
`initConfiguration`	`object`	initConfiguration along with ClusterConfiguration are the configurations necessary for the init command

Property	Type	Description
`joinConfiguration`	`object`	joinConfiguration is the kubeadm configuration for the join command
`mounts`	`array`	mounts specifies a list of mount points to be setup.
`ntp`	`object`	ntp specifies NTP configuration
`postKubeadmCommands`	`array`	postKubeadmCommands specifies extra commands to run after kubeadm runs. With cloud-init, this is appended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is appended to /etc/kubeadm.sh.
`preKubeadmCommands`	`array`	preKubeadmCommands specifies extra commands to run before kubeadm runs. With cloud-init, this is prepended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is prepended to /etc/kubeadm.sh.
`useExperimentalRetryJoin`	`boolean`	useExperimentalRetryJoin replaces a basic kubeadm command with a shell script with retries for joins. This is meant to be an experimental temporary workaround on some environments where joins fail due to timing (and other issues). The long term goal is to add retries to kubeadm proper and use that functionality. This will add about 40KB to userdata For more information, refer to https://github.com/kubernetes-sigs/cluster-api/pull/2763#discussion_r397306055↗. Deprecated: This experimental fix is no longer needed and this field will be removed in a future release. When removing also remove from staticcheck exclude-rules for SA1019 in golangci.yml
`users`	`array`	users specifies extra users to add
`verbosity`	`integer`	verbosity is the number for the kubeadm log level verbosity. It overrides the `--v` flag in kubeadm commands.

.spec.template.spec.kubeadmConfigSpec.bootCommands

Description

bootCommands specifies extra commands to run very early in the boot process via the cloud-init bootcmd module. bootcmd will run on every boot, 'cloud-init-per' command can be used to make bootcmd run exactly once. This is typically run in the cloud-init.service systemd unit. This has no effect in Ignition.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.bootCommands[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration

Description

clusterConfiguration along with InitConfiguration are the configurations necessary for the init command

Type

`object`

Property	Type	Description
`apiServer`	`object`	apiServer contains extra settings for the API server control plane component
`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 ↗
`certificatesDir`	`string`	certificatesDir specifies where to store or look for all required certificates. NB: if not provided, this will default to `/etc/kubernetes/pki`
`clusterName`	`string`	clusterName is the cluster name
`controlPlaneEndpoint`	`string`	controlPlaneEndpoint sets a stable IP address or DNS name for the control plane; it can be a valid IP address or a RFC-1123 DNS subdomain, both with optional TCP port. In case the ControlPlaneEndpoint is not specified, the AdvertiseAddress + BindPort are used; in case the ControlPlaneEndpoint is specified but without a TCP port, the BindPort is used. Possible usages are: e.g. In a cluster with more than one control plane instances, this field should be assigned the address of the external load balancer in front of the control plane instances. e.g. in environments with enforced node recycling, the ControlPlaneEndpoint could be used for assigning a stable DNS to the control plane. NB: This value defaults to the first value in the Cluster object status.apiEndpoints array.
`controllerManager`	`object`	controllerManager contains extra settings for the controller manager control plane component
`dns`	`object`	dns defines the options for the DNS add-on installed in the cluster.
`etcd`	`object`	etcd holds configuration for etcd. NB: This value defaults to a Local (stacked) etcd
`featureGates`	`object`	featureGates enabled by the user.

Property	Type	Description

`imageRepository`	`string`	imageRepository sets the container registry to pull images from.
		• If not set, the default registry of kubeadm will be used, i.e.
		  • registry.k8s.io (new registry): >= v1.22.17, >= v1.23.15, >= v1.24.9, >= v1.25.0
		  • k8s.gcr.io (old registry): all older versions Please note that when imageRepository is not set we don't allow upgrades to versions >= v1.22.0 which use the old registry (k8s.gcr.io). Please use a newer patch version with the new registry instead (i.e. >= v1.22.17,
		> = v1.23.15, >= v1.24.9, >= v1.25.0).
		• If the version is a CI build (kubernetes version starts with `ci/` or `ci-cross/`) `gcr.io/k8s-staging-ci-images` will be used as a default for control plane components and for kube-proxy, while `registry.k8s.io` will be used for all the other images.

`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 ↗
`kubernetesVersion`	`string`	kubernetesVersion is the target version of the control plane. NB: This value defaults to the Machine object spec.version
`networking`	`object`	networking holds configuration for the networking topology of the cluster. NB: This value defaults to the Cluster object spec.clusterNetwork.
`scheduler`	`object`	scheduler contains extra settings for the scheduler control plane component

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer

Description

apiServer contains extra settings for the API server control plane component

Type

`object`

Property	Type	Description
`certSANs`	`array`	certSANs sets extra Subject Alternative Names for the API Server signing cert.
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy

Property	Type	Description
		environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.
`timeoutForControlPlane`	`string`	timeoutForControlPlane controls the timeout that we use for API server to appear

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.certSANs

Description

certSANs sets extra Subject Alternative Names for the API Server signing cert.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.certSANs[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[]

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.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].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.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager

Description

controllerManager contains extra settings for the controller manager control plane component

Type

`object`

Property	Type	Description
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[]

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.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom

Description

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

Type

`object`

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

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].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.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key of the secret to select from. Must be a valid secret key.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.controllerManager.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.dns

Description

dns defines the options for the DNS add-on installed in the cluster.

Type

`object`

Property	Type	Description
`imageRepository`	`string`	imageRepository sets the container registry to pull images from. if not set, the ImageRepository defined in ClusterConfiguration will be used instead.
`imageTag`	`string`	imageTag allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd

Description

etcd holds configuration for etcd. NB: This value defaults to a Local (stacked) etcd

Type

`object`

Property	Type	Description
`external`	`object`	external describes how to connect to an external etcd cluster Local and External are mutually exclusive
`local`	`object`	local provides configuration knobs for configuring the local etcd instance Local and External are mutually exclusive

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.external

Description

external describes how to connect to an external etcd cluster Local and External are mutually exclusive

Type

`object`

Required

`caFile` `certFile` `endpoints` `keyFile`

Property	Type	Description
`caFile`	`string`	caFile is an SSL Certificate Authority file used to secure etcd communication. Required if using a TLS connection.
`certFile`	`string`	certFile is an SSL certification file used to secure etcd communication. Required if using a TLS connection.
`endpoints`	`array`	endpoints of etcd members. Required for ExternalEtcd.
`keyFile`	`string`	keyFile is an SSL key file used to secure etcd communication. Required if using a TLS connection.

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.external.endpoints

Description

endpoints of etcd members. Required for ExternalEtcd.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.external.endpoints[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local

Description

local provides configuration knobs for configuring the local etcd instance Local and External are mutually exclusive

Type

`object`

Property	Type	Description
`dataDir`	`string`	dataDir is the directory etcd will place its data. Defaults to "/var/lib/etcd".
`extraArgs`	`object`	extraArgs are extra arguments provided to the etcd binary when run inside a static pod.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`imageRepository`	`string`	imageRepository sets the container registry to pull images from. if not set, the ImageRepository defined in ClusterConfiguration will be used instead.
`imageTag`	`string`	imageTag allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.
`peerCertSANs`	`array`	peerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.
`serverCertSANs`	`array`	serverCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraArgs

Description

extraArgs are extra arguments provided to the etcd binary when run inside a static pod.

Type

`object`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[]

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.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom

Description

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

Type

`object`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].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.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.peerCertSANs

Description

peerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.peerCertSANs[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.serverCertSANS

Description

serverCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.etcd.local.serverCertSANs[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.featureGates

Description

featureGates enabled by the user.

Type

`object`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.networking

Description

networking holds configuration for the networking topology of the cluster. NB: This value defaults to the Cluster object spec.clusterNetwork.

Type

`object`

Property	Type	Description
`dnsDomain`	`string`	dnsDomain is the dns domain used by k8s services. Defaults to "cluster.local".
`podSubnet`	`string`	podSubnet is the subnet used by pods. If unset, the API server will not allocate CIDR ranges for every node. Defaults to a comma-delimited string of the Cluster object's spec.clusterNetwork.services.cidrBlocks if that is set
`serviceSubnet`	`string`	serviceSubnet is the subnet used by k8s services. Defaults to a comma-delimited string of the Cluster object's spec.clusterNetwork.pods.cidrBlocks, or to "10.96.0.0/12" if that's unset.

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler

Description

scheduler contains extra settings for the scheduler control plane component

Type

`object`

Property	Type	Description
`extraArgs`	`object`	extraArgs is an extra set of flags to pass to the control plane component.
`extraEnvs`	`array`	extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.
`extraVolumes`	`array`	extraVolumes is an extra set of host volumes, mounted to the control plane component.

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraArgs

Description

extraArgs is an extra set of flags to pass to the control plane component.

Type

`object`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs

Description

extraEnvs is an extra set of environment variables to pass to the control plane component. Environment variables passed using ExtraEnvs will override any existing environment variables, or *_proxy environment variables that kubeadm adds by default. This option takes effect only on Kubernetes >=1.31.0.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[]

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.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].valueFrom

Description

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

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`fieldRef`	`object`	Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
`secretKeyRef`	`object`	Selects a key of a secret in the pod's namespace

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].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.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.resourceFieldRef

Description

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

Type

`object`

Required

`resource`

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

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraEnvs[].valueFrom.secretKeyRef

Description

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

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key of the secret to select from. Must be a valid secret key.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraVolumes

Description

extraVolumes is an extra set of host volumes, mounted to the control plane component.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.clusterConfiguration.scheduler.extraVolumes[]

Description

HostPathMount contains elements describing volumes that are mounted from the host.

Type

`object`

Required

`hostPath` `mountPath` `name`

Property	Type	Description
`hostPath`	`string`	hostPath is the path in the host that will be mounted inside the pod.
`mountPath`	`string`	mountPath is the path inside the pod where hostPath will be mounted.
`name`	`string`	name of the volume inside the pod template.
`pathType`	`string`	pathType is the type of the HostPath.
`readOnly`	`boolean`	readOnly controls write access to the volume

.spec.template.spec.kubeadmConfigSpec.diskSetup

Description

diskSetup specifies options for the creation of partition tables and file systems on devices.

Type

`object`

Property	Type	Description
`filesystems`	`array`	filesystems specifies the list of file systems to setup.
`partitions`	`array`	partitions specifies the list of the partitions to setup.

.spec.template.spec.kubeadmConfigSpec.diskSetup.filesystems

Description

filesystems specifies the list of file systems to setup.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.diskSetup.filesystems[]

Description

Filesystem defines the file systems to be created.

Type

`object`

Required

`device` `filesystem`

Property	Type	Description
`device`	`string`	device specifies the device name
`extraOpts`	`array`	extraOpts defined extra options to add to the command for creating the file system.
`filesystem`	`string`	filesystem specifies the file system type.
`label`	`string`	label specifies the file system label to be used. If set to None, no label is used.
`overwrite`	`boolean`	overwrite defines whether or not to overwrite any existing filesystem. If true, any pre-existing file system will be destroyed. Use with Caution.
`partition`	`string`	partition specifies the partition to use. The valid options are: "auto|any", "auto", "any", "none", and , where NUM is the actual partition number.

Property	Type	Description
`replaceFS`	`string`	replaceFS is a special directive, used for Microsoft Azure that instructs cloud-init to replace a file system of <FS_TYPE>. NOTE: unless you define a label, this requires the use of the 'any' partition directive.

.spec.template.spec.kubeadmConfigSpec.diskSetup.filesystems[].extraOpts

Description

extraOpts defined extra options to add to the command for creating the file system.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.diskSetup.filesystems[].extraOpts[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.diskSetup.partitions

Description

partitions specifies the list of the partitions to setup.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.diskSetup.partitions[]

Description

Partition defines how to create and layout a partition.

Type

`object`

Required

`device` `layout`

Property	Type	Description
`device`	`string`	device is the name of the device.
`layout`	`boolean`	layout specifies the device layout. If it is true, a single partition will be created for the entire device. When layout is false, it means don't partition or ignore existing partitioning.
`overwrite`	`boolean`	overwrite describes whether to skip checks and create the partition if a partition or filesystem is found on the device. Use with caution. Default is 'false'.

Property	Type	Description
`tableType`	`string`	tableType specifies the tupe of partition table. The following are supported: 'mbr': default and setups a MS-DOS partition table 'gpt': setups a GPT partition table

.spec.template.spec.kubeadmConfigSpec.files

Description

files specifies extra files to be passed to user_data upon creation.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.files[]

Description

File defines the input for generating write_files in cloud-init.

Type

`object`

Required

`path`

Property	Type	Description
`append`	`boolean`	append specifies whether to append Content to existing file if Path exists.
`content`	`string`	content is the actual content of the file.
`contentFrom`	`object`	contentFrom is a referenced source of content to populate the file.
`encoding`	`string`	encoding specifies the encoding of the file contents.
`owner`	`string`	owner specifies the ownership of the file, e.g. "root:root".
`path`	`string`	path specifies the full path on disk where to store the file.
`permissions`	`string`	permissions specifies the permissions to assign to the file, e.g. "0640".

.spec.template.spec.kubeadmConfigSpec.files[].contentFrom

Description

contentFrom is a referenced source of content to populate the file.

Type

`object`

Required

`secret`

Property	Type	Description
`secret`	`object`	secret represents a secret that should populate this file.

.spec.template.spec.kubeadmConfigSpec.files[].contentFrom.secret

Description

secret represents a secret that should populate this file.

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	key is the key in the secret's data map for this value.
`name`	`string`	name of the secret in the KubeadmBootstrapConfig's namespace to use.

.spec.template.spec.kubeadmConfigSpec.ignition

Description

ignition contains Ignition specific configuration.

Type

`object`

Property	Type	Description
`containerLinuxConfig`	`object`	containerLinuxConfig contains CLC specific configuration.

.spec.template.spec.kubeadmConfigSpec.ignition.containerLinuxConfig

Description

containerLinuxConfig contains CLC specific configuration.

Type

`object`

Property	Type	Description
`additionalConfig`	`string`	additionalConfig contains additional configuration to be merged with the Ignition configuration generated by the bootstrapper controller. More info: https://coreos.github.io/ignition/operator-notes/#config-merging ↗ The data format is documented here: https://kinvolk.io/docs/flatcar-container-linux/latest/provisioning/cl-config/ ↗
`strict`	`boolean`	strict controls if AdditionalConfig should be strictly parsed. If so, warnings are treated as errors.

.spec.template.spec.kubeadmConfigSpec.initConfiguration

Description

initConfiguration along with ClusterConfiguration are the configurations necessary for the init command

Type

`object`

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 ↗
`bootstrapTokens`	`array`	bootstrapTokens is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature
`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 ↗
`localAPIEndpoint`	`object`	localAPIEndpoint represents the endpoint of the API server instance that's deployed on this control plane node In HA setups, this differs from ClusterConfiguration.ControlPlaneEndpoint in the sense that ControlPlaneEndpoint is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.
`nodeRegistration`	`object`	nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Property	Type	Description
`patches`	`object`	patches contains options related to applying patches to components deployed by kubeadm during "kubeadm init". The minimum kubernetes version needed to support Patches is v1.22
`skipPhases`	`array`	skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

.spec.template.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens

Description

bootstrapTokens is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature

Type

`array`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[]

Description

BootstrapToken describes one bootstrap token, stored as a Secret in the cluster.

Type

`object`

Required

`token`

Property	Type	Description
`description`	`string`	description sets a human-friendly message why this token exists and what it's used for, so other administrators can know its purpose.
`expires`	`string`	expires specifies the timestamp when this token expires. Defaults to being set dynamically at runtime based on the TTL. Expires and TTL are mutually exclusive.
`groups`	`array`	groups specifies the extra groups that this token will authenticate as when/if used for authentication
`token`	`string`	token is used for establishing bidirectional trust between nodes and control-planes. Used for joining nodes in the cluster.
`ttl`	`string`	ttl defines the time to live for this token. Defaults to 24h. Expires and TTL are mutually exclusive.

Property	Type	Description
`usages`	`array`	usages describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.

.spec.template.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[].groups

Description

groups specifies the extra groups that this token will authenticate as when/if used for authentication

Type

`array`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[].groups[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[].usages

Description

usages describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.bootstrapTokens[].usages[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.localAPIEndpoint

Description

localAPIEndpoint represents the endpoint of the API server instance that's deployed on this control plane node In HA setups, this differs from ClusterConfiguration.ControlPlaneEndpoint in the sense that ControlPlaneEndpoint is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.

Type

`object`

Property	Type	Description
`advertiseAddress`	`string`	advertiseAddress sets the IP address for the API server to advertise.

Property	Type	Description
`bindPort`	`integer`	bindPort sets the secure port for the API Server to bind to. Defaults to 6443.

.spec.template.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration

Description

nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Type

`object`

Property	Type	Description
`criSocket`	`string`	criSocket is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use
`ignorePreflightErrors`	`array`	ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.
`imagePullPolicy`	`string`	imagePullPolicy specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". Defaults to "IfNotPresent". This can be used only with Kubernetes version equal to 1.22 and later.
`imagePullSerial`	`boolean`	imagePullSerial specifies if image pulling performed by kubeadm must be done serially or in parallel. This option takes effect only on Kubernetes >=1.31.0. Default: true (defaulted in kubeadm)
`kubeletExtraArgs`	`object`	kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.
`name`	`string`	name is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the CommonName field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`taints`	`array`	taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

.spec.template.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.ignorePreflightErrors

Description

ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.ignorePreflightErrors[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.kubeletExtraArgs

Description

kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.

Type

`object`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.taints

Description

taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.

Property	Type	Description
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.template.spec.kubeadmConfigSpec.initConfiguration.patches

Description

patches contains options related to applying patches to components deployed by kubeadm during "kubeadm init". The minimum kubernetes version needed to support Patches is v1.22

Type

`object`

Property	Type	Description
`directory`	`string`	directory is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic" "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically. These files can be written into the target directory via KubeadmConfig.Files which specifies additional files to be created on the machine, either with content inline or by referencing a secret.

.spec.template.spec.kubeadmConfigSpec.initConfiguration.skipPhases

Description

skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.initConfiguration.skipPhases[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration

Description

joinConfiguration is the kubeadm configuration for the join command

Type

`object`

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 ↗
`caCertPath`	`string`	caCertPath is the path to the SSL certificate authority used to secure comunications between node and control-plane. Defaults to "/etc/kubernetes/pki/ca.crt".
`controlPlane`	`object`	controlPlane defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.
`discovery`	`object`	discovery specifies the options for the kubelet to use during the TLS Bootstrap process
`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 ↗
`nodeRegistration`	`object`	nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration
`patches`	`object`	patches contains options related to applying patches to components deployed by kubeadm during "kubeadm join". The minimum kubernetes version needed to support Patches is v1.22
`skipPhases`	`array`	skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.controlPlane

Description

controlPlane defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.

Type

`object`

Property	Type	Description
`localAPIEndpoint`	`object`	localAPIEndpoint represents the endpoint of the API server instance to be deployed on this node.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.controlPlane.localAPIEndpoint

Description

localAPIEndpoint represents the endpoint of the API server instance to be deployed on this node.

Type

`object`

Property	Type	Description
`advertiseAddress`	`string`	advertiseAddress sets the IP address for the API server to advertise.
`bindPort`	`integer`	bindPort sets the secure port for the API Server to bind to. Defaults to 6443.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery

Description

discovery specifies the options for the kubelet to use during the TLS Bootstrap process

Type

`object`

Property	Type	Description
`bootstrapToken`	`object`	bootstrapToken is used to set the options for bootstrap token based discovery BootstrapToken and File are mutually exclusive
`file`	`object`	file is used to specify a file or URL to a kubeconfig file from which to load cluster information BootstrapToken and File are mutually exclusive
`timeout`	`string`	timeout modifies the discovery timeout
`tlsBootstrapToken`	`string`	tlsBootstrapToken is a token used for TLS bootstrapping. If .BootstrapToken is set, this field is defaulted to .BootstrapToken.Token, but can be overridden. If .File is set, this field **must be set** in case the KubeConfigFile does not contain any other authentication information

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.bootstrapToken

Description

bootstrapToken is used to set the options for bootstrap token based discovery BootstrapToken and File are mutually exclusive

Type

`object`

Property	Type	Description
`apiServerEndpoint`	`string`	apiServerEndpoint is an IP or domain name to the API server from which info will be fetched.
`caCertHashes`	`array`	caCertHashes specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as ":", where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL: openssl x509 -pubkey -in ca.crt openssl rsa -pubin -outform der 2>&/dev/null | openssl dgst -sha256 -hex
`token`	`string`	token is a token used to validate cluster information fetched from the control-plane.
`unsafeSkipCAVerification`	`boolean`	unsafeSkipCAVerification allows token-based discovery without CA verification via CACertHashes. This can weaken the security of kubeadm since other nodes can impersonate the control-plane.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.bootstrapToken.caCertHashes

Description

caCertHashes specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as "<type>:<value>", where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL: openssl x509 -pubkey -in ca.crt openssl rsa -pubin -outform der 2>&/dev/null | openssl dgst -sha256 -hex

Type

`array`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.bootstrapToken.caCertHashes[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file

Description

file is used to specify a file or URL to a kubeconfig file from which to load cluster information BootstrapToken and File are mutually exclusive

Type

`object`

Required

`kubeConfigPath`

Property	Type	Description
`kubeConfig`	`object`	kubeConfig is used (optionally) to generate a KubeConfig based on the KubeadmConfig's information. The file is generated at the path specified in KubeConfigPath. Host address (server field) information is automatically populated based on the Cluster's ControlPlaneEndpoint. Certificate Authority (certificate-authority-data field) is gathered from the cluster's CA secret.
`kubeConfigPath`	`string`	kubeConfigPath is used to specify the actual file path or URL to the kubeconfig file from which to load cluster information

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig

Description

kubeConfig is used (optionally) to generate a KubeConfig based on the KubeadmConfig's information. The file is generated at the path specified in KubeConfigPath. Host address (server field) information is automatically populated based on the Cluster's ControlPlaneEndpoint. Certificate Authority (certificate-authority-data field) is gathered from the cluster's CA secret.

Type

`object`

Required

`user`

Property	Type	Description
`cluster`	`object`	cluster contains information about how to communicate with the kubernetes cluster. By default the following fields are automatically populated: • Server with the Cluster's ControlPlaneEndpoint. • CertificateAuthorityData with the Cluster's CA certificate.
`user`	`object`	user contains information that describes identity information. This is used to tell the kubernetes cluster who you are.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.cluster

Description

cluster contains information about how to communicate with the kubernetes cluster. By default the following fields are automatically populated: - Server with the Cluster's ControlPlaneEndpoint. - CertificateAuthorityData with the Cluster's CA certificate.

Type

`object`

Property	Type	Description
`certificateAuthorityData`	`string`	certificateAuthorityData contains PEM-encoded certificate authority certificates.

Property	Type	Description
		Defaults to the Cluster's CA certificate if empty.
`insecureSkipTLSVerify`	`boolean`	insecureSkipTLSVerify skips the validity check for the server's certificate. This will make your HTTPS connections insecure.
`proxyURL`	`string`	proxyURL is the URL to the proxy to be used for all requests made by this client. URLs with "http", "https", and "socks5" schemes are supported. If this configuration is not provided or the empty string, the client attempts to construct a proxy configuration from http_proxy and https_proxy environment variables. If these environment variables are not set, the client does not attempt to proxy requests. socks5 proxying does not currently support spdy streaming endpoints (exec, attach, port forward).
`server`	`string`	server is the address of the kubernetes cluster (https://hostname:port↗). Defaults to https:// + Cluster.Spec.ControlPlaneEndpoint.
`tlsServerName`	`string`	tlsServerName is used to check server certificate. If TLSServerName is empty, the hostname used to contact the server is used.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user

Description

user contains information that describes identity information. This is used to tell the kubernetes cluster who you are.

Type

`object`

Property	Type	Description
`authProvider`	`object`	authProvider specifies a custom authentication plugin for the kubernetes cluster.
`exec`	`object`	exec specifies a custom exec-based authentication plugin for the kubernetes cluster.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.authProvider

Description

authProvider specifies a custom authentication plugin for the kubernetes cluster.

Type

`object`

Required

`name`

Property	Type	Description
`config`	`object`	config holds the parameters for the authentication plugin.
`name`	`string`	name is the name of the authentication plugin.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.authProvider.config

Description

config holds the parameters for the authentication plugin.

Type

`object`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec

Description

exec specifies a custom exec-based authentication plugin for the kubernetes cluster.

Type

`object`

Required

`command`

Property	Type	Description
`apiVersion`	`string`	apiVersion is preferred input version of the ExecInfo. The returned ExecCredentials MUST use the same encoding version as the input. Defaults to client.authentication.k8s.io/v1 if not set.
`args`	`array`	args is the arguments to pass to the command when executing it.
`command`	`string`	command to execute.
`env`	`array`	env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

Property	Type	Description
`provideClusterInfo`	`boolean`	provideClusterInfo determines whether or not to provide cluster information, which could potentially contain very large CA data, to this exec plugin as a part of the KUBERNETES_EXEC_INFO environment variable. By default, it is set to false. Package k8s.io/client-go/tools/auth/exec provides helper methods for reading this environment variable.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec.args

Description

args is the arguments to pass to the command when executing it.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec.args[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec.env

Description

env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.discovery.file.kubeConfig.user.exec.env[]

Description

KubeConfigAuthExecEnv is used for setting environment variables when executing an exec-based credential plugin.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	name of the environment variable

Property	Type	Description
`value`	`string`	value of the environment variable

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration

Description

nodeRegistration holds fields that relate to registering the new control-plane node to the cluster. When used in the context of control plane nodes, NodeRegistration should remain consistent across both InitConfiguration and JoinConfiguration

Type

`object`

Property	Type	Description
`criSocket`	`string`	criSocket is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use
`ignorePreflightErrors`	`array`	ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.
`imagePullPolicy`	`string`	imagePullPolicy specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". Defaults to "IfNotPresent". This can be used only with Kubernetes version equal to 1.22 and later.
`imagePullSerial`	`boolean`	imagePullSerial specifies if image pulling performed by kubeadm must be done serially or in parallel. This option takes effect only on Kubernetes >=1.31.0. Default: true (defaulted in kubeadm)
`kubeletExtraArgs`	`object`	kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.
`name`	`string`	name is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the CommonName field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`taints`	`array`	taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.ignorePreflightErrors

Description

ignorePreflightErrors provides a slice of pre-flight errors to be ignored when the current node is registered.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.ignorePreflightErrors[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.kubeletExtraArgs

Description

kubeletExtraArgs passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the kubelet-config-1.X ConfigMap Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on.

Type

`object`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.taints

Description

taints specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, in the `kubeadm init` process it will be defaulted to []v1.Taint{'node-role.kubernetes.io/master=""}. If you don't want to taint your control-plane node, set this field to an empty slice, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.nodeRegistration.taints[]

Description

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

Type

`object`

Required

`effect` `key`

Property	Type	Description
`effect`	`string`	Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.

Property	Type	Description
`key`	`string`	Required. The taint key to be applied to a node.
`timeAdded`	`string`	TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.
`value`	`string`	The taint value corresponding to the taint key.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.patches

Description

patches contains options related to applying patches to components deployed by kubeadm during "kubeadm join". The minimum kubernetes version needed to support Patches is v1.22

Type

`object`

Property	Type	Description
`directory`	`string`	directory is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic" "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically. These files can be written into the target directory via KubeadmConfig.Files which specifies additional files to be created on the machine, either with content inline or by referencing a secret.

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.skipPhases

Description

skipPhases is a list of phases to skip during command execution. The list of phases can be obtained with the "kubeadm init --help" command. This option takes effect only on Kubernetes >=1.22.0.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.joinConfiguration.skipPhases[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.mounts

Description

mounts specifies a list of mount points to be setup.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.mounts[]

Description

MountPoints defines input for generated mounts in cloud-init.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.ntp

Description

ntp specifies NTP configuration

Type

`object`

Property	Type	Description
`enabled`	`boolean`	enabled specifies whether NTP should be enabled
`servers`	`array`	servers specifies which NTP servers to use

.spec.template.spec.kubeadmConfigSpec.ntp.servers

Description

servers specifies which NTP servers to use

Type

`array`

.spec.template.spec.kubeadmConfigSpec.ntp.servers[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.postKubeadmCommands

Description

postKubeadmCommands specifies extra commands to run after kubeadm runs. With cloud-init, this is appended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is appended to /etc/kubeadm.sh.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.postKubeadmCommands[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.preKubeadmCommands

Description

preKubeadmCommands specifies extra commands to run before kubeadm runs. With cloud-init, this is prepended to the runcmd module configuration, and is typically executed in the cloud-final.service systemd unit. In Ignition, this is prepended to /etc/kubeadm.sh.

Type

`array`

.spec.template.spec.kubeadmConfigSpec.preKubeadmCommands[]

Type

`string`

.spec.template.spec.kubeadmConfigSpec.users

Description

users specifies extra users to add

Type

`array`

.spec.template.spec.kubeadmConfigSpec.users[]

Description

User defines the input for a generated user in cloud-init.

Type

`object`

Required

`name`

Property	Type	Description
`gecos`	`string`	gecos specifies the gecos to use for the user
`groups`	`string`	groups specifies the additional groups for the user
`homeDir`	`string`	homeDir specifies the home directory to use for the user
`inactive`	`boolean`	inactive specifies whether to mark the user as inactive
`lockPassword`	`boolean`	lockPassword specifies if password login should be disabled
`name`	`string`	name specifies the user name

Property	Type	Description
`passwd`	`string`	passwd specifies a hashed password for the user
`passwdFrom`	`object`	passwdFrom is a referenced source of passwd to populate the passwd.
`primaryGroup`	`string`	primaryGroup specifies the primary group for the user
`shell`	`string`	shell specifies the user's shell
`sshAuthorizedKeys`	`array`	sshAuthorizedKeys specifies a list of ssh authorized keys for the user
`sudo`	`string`	sudo specifies a sudo role for the user

.spec.template.spec.kubeadmConfigSpec.users[].passwdFrom

Description

passwdFrom is a referenced source of passwd to populate the passwd.

Type

`object`

Required

`secret`

Property	Type	Description
`secret`	`object`	secret represents a secret that should populate this password.

.spec.template.spec.kubeadmConfigSpec.users[].passwdFrom.secret

Description

secret represents a secret that should populate this password.

Type

`object`

Required

`key` `name`

Property	Type	Description
`key`	`string`	key is the key in the secret's data map for this value.

Property	Type	Description
`name`	`string`	name of the secret in the KubeadmBootstrapConfig's namespace to use.

.spec.template.spec.kubeadmConfigSpec.users[].sshAuthorizedKeys

Description

sshAuthorizedKeys specifies a list of ssh authorized keys for the user

Type

`array`

.spec.template.spec.kubeadmConfigSpec.users[].sshAuthorizedKeys[]

Type

`string`

.spec.template.spec.machineNamingStrategy

Description

machineNamingStrategy allows changing the naming pattern used when creating Machines. InfraMachines & KubeadmConfigs will use the same name as the corresponding Machines.

Type

`object`

Property	Type	Description
`template`	`string`	template defines the template to use for generating the names of the Machine objects. If not defined, it will fallback to `{{ .kubeadmControlPlane.name }}-{{ .random }}`. If the generated name string exceeds 63 characters, it will be trimmed to 58 characters and will get concatenated with a random suffix of length 5. Length of the template string must not exceed 256 characters. The template allows the following variables `.cluster.name`, `.kubeadmControlPlane.name` and `.random`. The variable `.cluster.name` retrieves the name of the cluster object that owns the Machines being created. The variable `.kubeadmControlPlane.name` retrieves the name of the KubeadmControlPlane object that owns the Machines being created. The variable `.random` is substituted with random alphanumeric string, without vowels, of length 5. This variable is required part of the template. If not provided, validation will fail.

.spec.template.spec.machineTemplate

Description

machineTemplate contains information about how machines should be shaped when creating or updating a control plane.

Type

`object`

Property	Type	Description
`metadata`	`ObjectMeta`	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
`nodeDeletionTimeout`	`string`	nodeDeletionTimeout defines how long the machine controller will attempt to delete the Node that the Machine hosts after the Machine is marked for deletion. A duration of 0 will retry deletion indefinitely. If no value is provided, the default value for this property of the Machine resource will be used.
`nodeDrainTimeout`	`string`	nodeDrainTimeout is the total amount of time that the controller will spend on draining a controlplane node The default value is 0, meaning that the node can be drained without any time limitations. NOTE: NodeDrainTimeout is different from `kubectl drain --timeout`
`nodeVolumeDetachTimeout`	`string`	nodeVolumeDetachTimeout is the total amount of time that the controller will spend on waiting for all volumes to be detached. The default value is 0, meaning that the volumes can be detached without any time limitations.

.spec.template.spec.remediationStrategy

Description

remediationStrategy is the RemediationStrategy that controls how control plane machine remediation happens.

Type

`object`

Property	Type	Description

`maxRetry`	`integer`	maxRetry is the Max number of retries while attempting to remediate an unhealthy machine. A retry happens when a created as a replacement for an unhealthy machine also fails. For example, given a control plane with three machine
		`M1 become unhealthy; remediation happens, and M1-1 is created as a replacement.`
		`If M1-1 (replacement of M1) has problems while bootstrapping it will become unhealthy`
		`remediated; such operation is considered a retry, remediation-retry #1.`
		`If M1-2 (replacement of M1-1) becomes unhealthy, remediation-retry #2 will happen, et`
		A retry could happen only after RetryPeriod from the previous retry. If a machine is marked as unhealthy after MinHe the previous remediation expired, this is not considered a retry anymore because the new issue is assumed unrelate one.
		If not set, the remedation will be retried infinitely.

`minHealthyPeriod`	`string`	minHealthyPeriod defines the duration after which KCP will consider any failure to a machine unrelated from the pre case the remediation is not considered a retry anymore, and thus the retry counter restarts from 0. For example, ass MinHealthyPeriod is set to 1h (default)

Property	Type	Description
		M1 become unhealthy; remediation happens, and M1-1 is created as a replacement. If M1-1 (replacement of M1) has problems within the 1hr after the creation, also this machine will be remediated and this operation is considered a retry - a problem to the original issue happened to M1 -. If instead the problem on M1-1 is happening after MinHealthyPeriod expired, e.g. four m1-1 has been created as a remediation of M1, the problem on M1-1 is considered unrel the original issue happened to M1. If not set, this value is defaulted to 1h.
`retryPeriod`	`string`	retryPeriod is the duration that KCP should wait before remediating a machine being created as a replacement for a machine (a retry). If not set, a retry will happen immediately.

.spec.template.spec.rolloutBefore

Description

rolloutBefore is a field to indicate a rollout should be performed if the specified criteria is met.

Type

`object`

Property	Type	Description
`certificatesExpiryDays`	`integer`	certificatesExpiryDays indicates a rollout needs to be performed if the certificates of the machine will expire within the specified days.

.spec.template.spec.rolloutStrategy

Description

rolloutStrategy is the RolloutStrategy to use to replace control plane machines with new ones.

Type

`object`

Property	Type	Description
`rollingUpdate`	`object`	rollingUpdate is the rolling update config params. Present only if RolloutStrategyType = RollingUpdate.
`type`	`string`	type of rollout. Currently the only supported strategy is "RollingUpdate". Default is RollingUpdate.

.spec.template.spec.rolloutStrategy.rollingUpdate

Description

rollingUpdate is the rolling update config params. Present only if RolloutStrategyType = RollingUpdate.

Type

`object`

Property	Type	Description
`maxSurge`	`integer|string`	maxSurge is the maximum number of control planes that can be scheduled above or under the desired number of control planes. Value can be an absolute number 1 or 0. Defaults to 1. Example: when this is set to 1, the control plane can be scaled up immediately when the rolling update starts.

API Endpoints

The following API endpoints are available:

- `/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanetemplates`
  - `DELETE` : delete collection of KubeadmControlPlaneTemplate
  - `GET` : list objects of kind KubeadmControlPlaneTemplate
  - `POST` : create a new KubeadmControlPlaneTemplate
- `/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanetemplates/{name}`
  - `DELETE` : delete the specified KubeadmControlPlaneTemplate
  - `GET` : read the specified KubeadmControlPlaneTemplate
  - `PATCH` : partially update the specified KubeadmControlPlaneTemplate
  - `PUT` : replace the specified KubeadmControlPlaneTemplate

/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanetemplates

HTTP method

`DELETE`

Description

delete collection of KubeadmControlPlaneTemplate

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind KubeadmControlPlaneTemplate

HTTP responses

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

HTTP method

`POST`

Description

create a new KubeadmControlPlaneTemplate

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`KubeadmControlPlaneTemplate` schema	`application/json` formatted

HTTP responses

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

/apis/controlplane.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/kubeadmcontrolplanetemplates/{name}

HTTP method

`DELETE`

Description

delete the specified KubeadmControlPlaneTemplate

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 KubeadmControlPlaneTemplate

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified KubeadmControlPlaneTemplate

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified KubeadmControlPlaneTemplate

Query parameters

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

Body parameters

Parameter	Type	Description
`body`	`KubeadmControlPlaneTemplate` schema	`application/json` formatted

HTTP responses

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