



OpenShift Container Platform 4.17

Machine APIs

Reference guide for machine APIs

OpenShift Container Platform 4.17 Machine APIs

Reference guide for machine APIs

Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

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

Table of Contents

CHAPTER 1. MACHINE APIS	7
1.1. CONTAINERRUNTIMECONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]	7
1.2. CONTROLLERCONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]	7
1.3. CONTROLPLANEMACHINESET [MACHINE.OPENSIFT.IO/V1]	7
1.4. KUBELETCONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]	7
1.5. MACHINECONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]	7
1.6. MACHINECONFIGPOOL [MACHINECONFIGURATION.OPENSIFT.IO/V1]	8
1.7. MACHINEHEALTHCHECK [MACHINE.OPENSIFT.IO/VIBETA1]	8
1.8. MACHINE [MACHINE.OPENSIFT.IO/VIBETA1]	8
1.9. MACHINESSET [MACHINE.OPENSIFT.IO/VIBETA1]	8
CHAPTER 2. CONTAINERRUNTIMECONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]	9
2.1. SPECIFICATION	9
2.1.1. .spec	10
2.1.2. .spec.containerRuntimeConfig	10
2.1.3. .spec.machineConfigPoolSelector	11
2.1.4. .spec.machineConfigPoolSelector.matchExpressions	12
2.1.5. .spec.machineConfigPoolSelector.matchExpressions[]	12
2.1.6. .status	12
2.1.7. .status.conditions	13
2.1.8. .status.conditions[]	13
2.2. API ENDPOINTS	14
2.2.1. /apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs	14
2.2.2. /apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs/{name}	16
2.2.3. /apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs/{name}/status	19
CHAPTER 3. CONTROLLERCONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]	23
3.1. SPECIFICATION	23
3.1.1. .spec	24
3.1.2. .spec.dns	26
3.1.3. .spec.imageRegistryBundleData	27
3.1.4. .spec.imageRegistryBundleData[]	27
3.1.5. .spec.imageRegistryBundleUserData	28
3.1.6. .spec.imageRegistryBundleUserData[]	28
3.1.7. .spec.infra	28
3.1.8. .spec.pullSecret	29
3.1.9. .status	31
3.1.10. .status.conditions	31
3.1.11. .status.conditions[]	31
3.1.12. .status.controllerCertificates	32
3.1.13. .status.controllerCertificates[]	32
3.2. API ENDPOINTS	33
3.2.1. /apis/machineconfiguration.openshift.io/v1/controllerconfigs	33
3.2.2. /apis/machineconfiguration.openshift.io/v1/controllerconfigs/{name}	35
3.2.3. /apis/machineconfiguration.openshift.io/v1/controllerconfigs/{name}/status	38
CHAPTER 4. CONTROLPLANEMACHINESET [MACHINE.OPENSIFT.IO/V1]	42
4.1. SPECIFICATION	42
4.1.1. .spec	43
4.1.2. .spec.selector	44
4.1.3. .spec.selector.matchExpressions	45
4.1.4. .spec.selector.matchExpressions[]	45

4.1.5. .spec.strategy	46
4.1.6. .spec.template	46
4.1.7. .spec.template.machines_v1beta1_machine_openshift_io	47
4.1.8. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains	48
4.1.9. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws	49
4.1.10. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[]	50
4.1.11. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[].placement	50
4.1.12. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[].subnet	50
4.1.13. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[].subnet.filters	51
4.1.14. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[].subnet.filters[]	51
4.1.15. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.azure	51
4.1.16. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.azure[]	52
4.1.17. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.gcp	52
4.1.18. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.gcp[]	52
4.1.19. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.nutanix	53
4.1.20. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.nutanix[]	53
4.1.21. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.openstack	53
4.1.22. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.openstack[]	53
4.1.23. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.openstack[].rootVolume	54
4.1.24. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.vsphere	55
4.1.25. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.vsphere[]	55
4.1.26. .spec.template.machines_v1beta1_machine_openshift_io.metadata	56
4.1.27. .spec.template.machines_v1beta1_machine_openshift_io.spec	57
4.1.28. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks	59
4.1.29. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks.preDrain	60
4.1.30. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks.preDrain[]	60
4.1.31. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks.preTerminate	60
4.1.32. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks.preTerminate[]	61
4.1.33. .spec.template.machines_v1beta1_machine_openshift_io.spec.metadata	61
4.1.34. .spec.template.machines_v1beta1_machine_openshift_io.spec.metadata.ownerReferences	63
4.1.35. .spec.template.machines_v1beta1_machine_openshift_io.spec.metadata.ownerReferences[]	64
4.1.36. .spec.template.machines_v1beta1_machine_openshift_io.spec.providerSpec	65
4.1.37. .spec.template.machines_v1beta1_machine_openshift_io.spec.taints	65
4.1.38. .spec.template.machines_v1beta1_machine_openshift_io.spec.taints[]	65
4.1.39. .status	66
4.1.40. .status.conditions	68
4.1.41. .status.conditions[]	68
4.2. API ENDPOINTS	70
4.2.1. /apis/machine.openshift.io/v1/controlplanemachinesets	71
4.2.2. /apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachinesets	71
4.2.3. /apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachinesets/{name}	73
4.2.4. /apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachinesets/{name}/scale	76
4.2.5. /apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachinesets/{name}/status	78
CHAPTER 5. KUBELETCONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]	82
5.1. SPECIFICATION	82
5.1.1. .spec	83
5.1.2. .spec.machineConfigPoolSelector	84
5.1.3. .spec.machineConfigPoolSelector.matchExpressions	84
5.1.4. .spec.machineConfigPoolSelector.matchExpressions[]	84
5.1.5. .spec.tlsSecurityProfile	85
5.1.6. .status	88
5.1.7. .status.conditions	88

5.1.8. .status.conditions[]	89
5.2. API ENDPOINTS	89
5.2.1. /apis/machineconfiguration.openshift.io/v1/kubeletconfigs	90
5.2.2. /apis/machineconfiguration.openshift.io/v1/kubeletconfigs/{name}	91
5.2.3. /apis/machineconfiguration.openshift.io/v1/kubeletconfigs/{name}/status	94
CHAPTER 6. MACHINECONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]	98
6.1. SPECIFICATION	98
6.1.1. .spec	98
6.2. API ENDPOINTS	99
6.2.1. /apis/machineconfiguration.openshift.io/v1/machineconfigs	100
6.2.2. /apis/machineconfiguration.openshift.io/v1/machineconfigs/{name}	101
CHAPTER 7. MACHINECONFIGPOOL [MACHINECONFIGURATION.OPENSIFT.IO/V1]	105
7.1. SPECIFICATION	105
7.1.1. .spec	106
7.1.2. .spec.configuration	107
7.1.3. .spec.configuration.source	110
7.1.4. .spec.configuration.source[]	110
7.1.5. .spec.machineConfigSelector	112
7.1.6. .spec.machineConfigSelector.matchExpressions	112
7.1.7. .spec.machineConfigSelector.matchExpressions[]	112
7.1.8. .spec.nodeSelector	113
7.1.9. .spec.nodeSelector.matchExpressions	114
7.1.10. .spec.nodeSelector.matchExpressions[]	114
7.1.11. .status	115
7.1.12. .status.certExpirys	116
7.1.13. .status.certExpirys[]	116
7.1.14. .status.conditions	117
7.1.15. .status.conditions[]	117
7.1.16. .status.configuration	118
7.1.17. .status.configuration.source	121
7.1.18. .status.configuration.source[]	121
7.2. API ENDPOINTS	123
7.2.1. /apis/machineconfiguration.openshift.io/v1/machineconfigpools	123
7.2.2. /apis/machineconfiguration.openshift.io/v1/machineconfigpools/{name}	125
7.2.3. /apis/machineconfiguration.openshift.io/v1/machineconfigpools/{name}/status	128
CHAPTER 8. MACHINEHEALTHCHECK [MACHINE.OPENSIFT.IO/V1BETA1]	132
8.1. SPECIFICATION	132
8.1.1. .spec	132
8.1.2. .spec.remediationTemplate	134
8.1.3. .spec.selector	136
8.1.4. .spec.selector.matchExpressions	136
8.1.5. .spec.selector.matchExpressions[]	136
8.1.6. .spec.unhealthyConditions	137
8.1.7. .spec.unhealthyConditions[]	137
8.1.8. .status	138
8.1.9. .status.conditions	138
8.1.10. .status.conditions[]	139
8.2. API ENDPOINTS	140
8.2.1. /apis/machine.openshift.io/v1beta1/machinehealthchecks	140
8.2.2. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinehealthchecks	141
8.2.3. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinehealthchecks/{name}	142

8.2.4. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinehealthchecks/{name}/status	145
CHAPTER 9. MACHINE [MACHINE.OPENSIFT.IO/V1BETA1]	149
9.1. SPECIFICATION	149
9.1.1. .spec	149
9.1.2. .spec.lifecycleHooks	151
9.1.3. .spec.lifecycleHooks.preDrain	152
9.1.4. .spec.lifecycleHooks.preDrain[]	152
9.1.5. .spec.lifecycleHooks.preTerminate	152
9.1.6. .spec.lifecycleHooks.preTerminate[]	153
9.1.7. .spec.metadata	153
9.1.8. .spec.metadata.ownerReferences	155
9.1.9. .spec.metadata.ownerReferences[]	156
9.1.10. .spec.providerSpec	157
9.1.11. .spec.taints	157
9.1.12. .spec.taints[]	157
9.1.13. .status	158
9.1.14. .status.addresses	161
9.1.15. .status.addresses[]	161
9.1.16. .status.conditions	161
9.1.17. .status.conditions[]	161
9.1.18. .status.lastOperation	163
9.1.19. .status.nodeRef	163
9.2. API ENDPOINTS	165
9.2.1. /apis/machine.openshift.io/v1beta1/machines	165
9.2.2. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machines	166
9.2.3. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machines/{name}	167
9.2.4. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machines/{name}/status	170
CHAPTER 10. MACHINESSET [MACHINE.OPENSIFT.IO/V1BETA1]	174
10.1. SPECIFICATION	174
10.1.1. .spec	175
10.1.2. .spec.selector	175
10.1.3. .spec.selector.matchExpressions	176
10.1.4. .spec.selector.matchExpressions[]	176
10.1.5. .spec.template	177
10.1.6. .spec.template.metadata	177
10.1.7. .spec.template.metadata.ownerReferences	179
10.1.8. .spec.template.metadata.ownerReferences[]	180
10.1.9. .spec.template.spec	181
10.1.10. .spec.template.spec.lifecycleHooks	183
10.1.11. .spec.template.spec.lifecycleHooks.preDrain	183
10.1.12. .spec.template.spec.lifecycleHooks.preDrain[]	183
10.1.13. .spec.template.spec.lifecycleHooks.preTerminate	184
10.1.14. .spec.template.spec.lifecycleHooks.preTerminate[]	184
10.1.15. .spec.template.spec.metadata	185
10.1.16. .spec.template.spec.metadata.ownerReferences	187
10.1.17. .spec.template.spec.metadata.ownerReferences[]	187
10.1.18. .spec.template.spec.providerSpec	189
10.1.19. .spec.template.spec.taints	189
10.1.20. .spec.template.spec.taints[]	189
10.1.21. .status	190

10.1.22. .status.conditions	192
10.1.23. .status.conditions[]	192
10.2. API ENDPOINTS	193
10.2.1. /apis/machine.openshift.io/v1beta1/machinesets	193
10.2.2. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets	194
10.2.3. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets/{name}	195
10.2.4. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets/{name}/scale	198
10.2.5. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets/{name}/status	201

CHAPTER 1. MACHINE APIS

1.1. CONTAINERRUNTIMECONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

ContainerRuntimeConfig describes a customized Container Runtime configuration. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.2. CONTROLLERCONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

ControllerConfig describes configuration for MachineConfigController. This is currently only used to drive the MachineConfig objects generated by the TemplateController. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.3. CONTROLPLANEMACHINESET [MACHINE.OPENSIFT.IO/V1]

Description

ControlPlaneMachineSet ensures that a specified number of control plane machine replicas are running at any given time. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.4. KUBELETCONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

KubeletConfig describes a customized Kubelet configuration. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.5. MACHINECONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

MachineConfig defines the configuration for a machine. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.6. MACHINECONFIGPOOL [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

MachineConfigPool describes a pool of MachineConfigs. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.7. MACHINEHEALTHCHECK [MACHINE.OPENSIFT.IO/V1BETA1]

Description

MachineHealthCheck is the Schema for the machinehealthchecks API Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

1.8. MACHINE [MACHINE.OPENSIFT.IO/V1BETA1]

Description

Machine is the Schema for the machines API Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

1.9. MACHINESET [MACHINE.OPENSIFT.IO/V1BETA1]

Description

MachineSet ensures that a specified number of machines replicas are running at any given time. Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

CHAPTER 2. CONTAINERRUNTIMECONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

ContainerRuntimeConfig describes a customized Container Runtime configuration. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

2.1. SPECIFICATION

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

Property	Type	Description
status	object	ContainerRuntimeConfigStatus defines the observed state of a ContainerRuntimeConfig

2.1.1. .spec

Description

ContainerRuntimeConfigSpec defines the desired state of ContainerRuntimeConfig

Type

object

Required

- **containerRuntimeConfig**

Property	Type	Description
containerRuntimeConfig	object	ContainerRuntimeConfiguration defines the tuneables of the container runtime
machineConfigPoolSelector	object	MachineConfigPoolSelector selects which pools the ContainerRuntimeConfig should apply to. A nil selector will result in no pools being selected.

2.1.2. .spec.containerRuntimeConfig

Description

ContainerRuntimeConfiguration defines the tuneables of the container runtime

Type

object

Property	Type	Description
defaultRuntime	string	defaultRuntime is the name of the OCI runtime to be used as the default.
logLevel	string	logLevel specifies the verbosity of the logs based on the level it is set to. Options are fatal, panic, error, warn, info, and debug.

Property	Type	Description
logSizeMax	integer-or-string	logSizeMax specifies the Maximum size allowed for the container log file. Negative numbers indicate that no size limit is imposed. If it is positive, it must be ≥ 8192 to match/exceed common's read buffer.
overlaySize	integer-or-string	overlaySize specifies the maximum size of a container image. This flag can be used to set quota on the size of container images. (default: 10GB)
pidsLimit	integer	pidsLimit specifies the maximum number of processes allowed in a container

2.1.3. .spec.machineConfigPoolSelector

Description

MachineConfigPoolSelector selects which pools the ContainerRuntimeConfig should apply to. A nil selector will result in no pools being selected.

Type

object

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

2.1.4. .spec.machineConfigPoolSelector.matchExpressions

Description

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

Type

array

2.1.5. .spec.machineConfigPoolSelector.matchExpressions[]

Description

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

Type

object

Required

- **key**
- **operator**

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

2.1.6. .status

Description

ContainerRuntimeConfigStatus defines the observed state of a ContainerRuntimeConfig

Type

object

Property	Type	Description
conditions	array	conditions represents the latest available observations of current state.
conditions[]	object	ContainerRuntimeConfigCondition defines the state of the ContainerRuntimeConfig
observedGeneration	integer	observedGeneration represents the generation observed by the controller.

2.1.7. .status.conditions

Description

conditions represents the latest available observations of current state.

Type

array

2.1.8. .status.conditions[]

Description

ContainerRuntimeConfigCondition defines the state of the ContainerRuntimeConfig

Type

object

Property	Type	Description
lastTransitionTime	string	lastTransitionTime is the time of the last update to the current status object.
message	string	message provides additional information about the current condition. This is only to be consumed by humans.
reason	string	reason is the reason for the condition's last transition. Reasons are PascalCase
status	string	status of the condition, one of True, False, Unknown.

Property	Type	Description
type	string	type specifies the state of the operator's reconciliation functionality.

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs**
 - **DELETE:** delete collection of ContainerRuntimeConfig
 - **GET:** list objects of kind ContainerRuntimeConfig
 - **POST:** create a ContainerRuntimeConfig
- **/apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs/{name}**
 - **DELETE:** delete a ContainerRuntimeConfig
 - **GET:** read the specified ContainerRuntimeConfig
 - **PATCH:** partially update the specified ContainerRuntimeConfig
 - **PUT:** replace the specified ContainerRuntimeConfig
- **/apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs/{name}/status**
 - **GET:** read status of the specified ContainerRuntimeConfig
 - **PATCH:** partially update status of the specified ContainerRuntimeConfig
 - **PUT:** replace status of the specified ContainerRuntimeConfig

2.2.1. /apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs

HTTP method

DELETE

Description

delete collection of ContainerRuntimeConfig

Table 2.1. HTTP responses

HTTP code	Response body
200 - OK	Status schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method**GET****Description**

list objects of kind ContainerRuntimeConfig

Table 2.2. HTTP responses

HTTP code	Reponse body
200 - OK	ContainerRuntimeConfigList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a ContainerRuntimeConfig

Table 2.3. Query parameters

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

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: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.4. Body parameters

Parameter	Type	Description
body	ContainerRuntimeConfig schema	

Table 2.5. HTTP responses

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

2.2.2. /apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs/{name}

Table 2.6. Global path parameters

Parameter	Type	Description
name	string	name of the ContainerRuntimeConfig

HTTP method**DELETE****Description**

delete a ContainerRuntimeConfig

Table 2.7. Query parameters

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

Table 2.8. HTTP responses

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

HTTP method**GET****Description**

read the specified ContainerRuntimeConfig

Table 2.9. HTTP responses

HTTP code	Reponse body
200 - OK	ContainerRuntimeConfig schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified ContainerRuntimeConfig

Table 2.10. Query parameters

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

Table 2.11. HTTP responses

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

HTTP method**PUT****Description**

replace the specified ContainerRuntimeConfig

Table 2.12. Query parameters

Parameter	Type	Description
-----------	------	-------------

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

Table 2.13. Body parameters

Parameter	Type	Description
body	ContainerRuntimeConfig schema	

Table 2.14. HTTP responses

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

2.2.3. /apis/machineconfiguration.openshift.io/v1/containerruntimeconfigs/{name}/s

Table 2.15. Global path parameters

Parameter	Type	Description
name	string	name of the ContainerRuntimeConfig

HTTP method**GET****Description**

read status of the specified ContainerRuntimeConfig

Table 2.16. HTTP responses

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

HTTP method**PATCH****Description**

partially update status of the specified ContainerRuntimeConfig

Table 2.17. Query parameters

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

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

Table 2.18. HTTP responses

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

HTTP method**PUT****Description**

replace status of the specified ContainerRuntimeConfig

Table 2.19. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.20. Body parameters

Parameter	Type	Description
body	ContainerRuntimeConfig schema	

Table 2.21. HTTP responses

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

CHAPTER 3. CONTROLLERCONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

ControllerConfig describes configuration for MachineConfigController. This is currently only used to drive the MachineConfig objects generated by the TemplateController. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

3.1. SPECIFICATION

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

Property	Type	Description
status	object	ControllerConfigStatus is the status for ControllerConfig

3.1.1. .spec

Description

ControllerConfigSpec is the spec for ControllerConfig resource.

Type

object

Required

- **baseOSContainerImage**
- **cloudProviderConfig**
- **clusterDNSIP**
- **images**
- **ipFamilies**
- **kubeAPIServerServingCAData**
- **releaseImage**
- **rootCAData**

Property	Type	Description
additionalTrustBundle	^^	additionalTrustBundle is a certificate bundle that will be added to the nodes trusted certificate store.
baseOSContainerImage	string	BaseOSContainerImage is the new-format container image for operating system updates.
baseOSExtensionsContainer Image	string	BaseOSExtensionsContainerImage is the matching extensions container for the new-format container
cloudProviderCAData	^^	cloudProvider specifies the cloud provider CA data

Property	Type	Description
cloudProviderConfig	string	cloudProviderConfig is the configuration for the given cloud provider
clusterDNSIP	string	clusterDNSIP is the cluster DNS IP address
dns	object	dns holds the cluster dns details
etcdDiscoveryDomain	string	etcdDiscoveryDomain is deprecated, use Infra.Status.EtcdDiscoveryDomain instead
imageRegistryBundleData	array	imageRegistryBundleData is the ImageRegistryData
imageRegistryBundleData[]	object	ImageRegistryBundle contains information for writing image registry certificates
imageRegistryBundleUserData	array	imageRegistryBundleUserData is Image Registry Data provided by the user
imageRegistryBundleUserData[]	object	ImageRegistryBundle contains information for writing image registry certificates
images	object (string)	images is map of images that are used by the controller to render templates under ./templates/
infra	object	infra holds the infrastructure details
internalRegistryPullSecret	''	internalRegistryPullSecret is the pull secret for the internal registry, used by rpm-ostree to pull images from the internal registry if present
ipFamilies	string	ipFamilies indicates the IP families in use by the cluster network

Property	Type	Description
kubeAPIServerServingCAData	string	kubeAPIServerServingCAData managed Kubelet to API Server Cert... Rotated automatically
network	``	Network contains additional network related information
networkType	string	networkType holds the type of network the cluster is using XXX: this is temporary and will be dropped as soon as possible in favor of a better support to start network related services the proper way. Nobody is also changing this once the cluster is up and running the first time, so, disallow regeneration if this changes.
osImageURL	string	OSImageURL is the old-format container image that contains the OS update payload.
platform	string	platform is deprecated, use Infra.Status.PlatformStatus.Type instead
proxy	``	proxy holds the current proxy configuration for the nodes
pullSecret	object	pullSecret is the default pull secret that needs to be installed on all machines.
releaseImage	string	releaseImage is the image used when installing the cluster
rootCAData	string	rootCAData specifies the root CA data

3.1.2. .spec.dns

Description

dns holds the cluster dns details

Type

object

Required

- **spec**
- **kind**
- **apiVersion**

Property	Type	Description
apiVersion	string	apiVersion defines the versioned schema of this representation of an object. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	kind is a string value representing the type of this object. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

3.1.3. .spec.imageRegistryBundleData**Description**

imageRegistryBundleData is the ImageRegistryData

Type

array

3.1.4. .spec.imageRegistryBundleData[]**Description**

ImageRegistryBundle contains information for writing image registry certificates

Type

object

Required

- **data**

- **file**

Property	Type	Description
data	string	data holds the contents of the bundle that will be written to the file location
file	string	file holds the name of the file where the bundle will be written to disk

3.1.5. .spec.imageRegistryBundleUserData

Description

imageRegistryBundleUserData is Image Registry Data provided by the user

Type

array

3.1.6. .spec.imageRegistryBundleUserData[]

Description

ImageRegistryBundle contains information for writing image registry certificates

Type

object

Required

- **data**
- **file**

Property	Type	Description
data	string	data holds the contents of the bundle that will be written to the file location
file	string	file holds the name of the file where the bundle will be written to disk

3.1.7. .spec.infra

Description

infra holds the infrastructure details

Type

object

Required

- **spec**
- **kind**
- **apiVersion**

Property	Type	Description
apiVersion	string	apiVersion defines the versioned schema of this representation of an object. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	kind is a string value representing the type of this object. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

3.1.8. .spec.pullSecret

Description

pullSecret is the default pull secret that needs to be installed on all machines.

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 <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: 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

3.1.9. .status

Description

ControllerConfigStatus is the status for ControllerConfig

Type

object

Property	Type	Description
conditions	array	conditions represents the latest available observations of current state.
conditions[]	object	ControllerConfigStatusCondition contains condition information for ControllerConfigStatus
controllerCertificates	array	controllerCertificates represents the latest available observations of the automatically rotating certificates in the MCO.
controllerCertificates[]	object	ControllerCertificate contains info about a specific cert.
observedGeneration	integer	observedGeneration represents the generation observed by the controller.

3.1.10. .status.conditions

Description

conditions represents the latest available observations of current state.

Type

array

3.1.11. .status.conditions[]

Description

ControllerConfigStatusCondition contains condition information for ControllerConfigStatus

Type**object****Required**

- **status**
- **type**

Property	Type	Description
lastTransitionTime	''	lastTransitionTime is the time of the last update to the current status object.
message	string	message provides additional information about the current condition. This is only to be consumed by humans.
reason	string	reason is the reason for the condition's last transition. Reasons are PascalCase
status	string	status of the condition, one of True, False, Unknown.
type	string	type specifies the state of the operator's reconciliation functionality.

3.1.12. .status.controllerCertificates

Description

controllerCertificates represents the latest available observations of the automatically rotating certificates in the MCO.

Type**array**

3.1.13. .status.controllerCertificates[]

Description

ControllerCertificate contains info about a specific cert.

Type**object****Required**

- **bundleFile**
- **signer**

- **subject**

Property	Type	Description
bundleFile	string	bundleFile is the larger bundle a cert comes from
notAfter	string	notAfter is the upper boundary for validity
notBefore	string	notBefore is the lower boundary for validity
signer	string	signer is the cert Issuer
subject	string	subject is the cert subject

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machineconfiguration.openshift.io/v1/controllerconfigs**
 - **DELETE**: delete collection of ControllerConfig
 - **GET**: list objects of kind ControllerConfig
 - **POST**: create a ControllerConfig
- **/apis/machineconfiguration.openshift.io/v1/controllerconfigs/{name}**
 - **DELETE**: delete a ControllerConfig
 - **GET**: read the specified ControllerConfig
 - **PATCH**: partially update the specified ControllerConfig
 - **PUT**: replace the specified ControllerConfig
- **/apis/machineconfiguration.openshift.io/v1/controllerconfigs/{name}/status**
 - **GET**: read status of the specified ControllerConfig
 - **PATCH**: partially update status of the specified ControllerConfig
 - **PUT**: replace status of the specified ControllerConfig

3.2.1. /apis/machineconfiguration.openshift.io/v1/controllerconfigs

HTTP method

DELETE

Description

delete collection of ControllerConfig

Table 3.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind ControllerConfig

Table 3.2. HTTP responses

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

HTTP method

POST

Description

create a ControllerConfig

Table 3.3. Query parameters

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

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

Table 3.4. Body parameters

Parameter	Type	Description
body	ControllerConfig schema	

Table 3.5. HTTP responses

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

3.2.2. /apis/machineconfiguration.openshift.io/v1/controllerconfigs/{name}

Table 3.6. Global path parameters

Parameter	Type	Description
name	string	name of the ControllerConfig

HTTP method**DELETE****Description**

delete a ControllerConfig

Table 3.7. Query parameters

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

Table 3.8. HTTP responses

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

HTTP method**GET****Description**

read the specified ControllerConfig

Table 3.9. HTTP responses

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

HTTP method**PATCH****Description**

partially update the specified ControllerConfig

Table 3.10. Query parameters

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

Table 3.11. HTTP responses

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

HTTP method**PUT****Description**

replace the specified ControllerConfig

Table 3.12. Query parameters

Parameter	Type	Description
-----------	------	-------------

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

Table 3.13. Body parameters

Parameter	Type	Description
body	ControllerConfig schema	

Table 3.14. HTTP responses

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

3.2.3. /apis/machineconfiguration.openshift.io/v1/controllerconfigs/{name}/status

Table 3.15. Global path parameters

Parameter	Type	Description
name	string	name of the ControllerConfig

HTTP method

GET

Description

read status of the specified ControllerConfig

Table 3.16. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified ControllerConfig

Table 3.17. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.18. HTTP responses

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

HTTP method**PUT****Description**

replace status of the specified ControllerConfig

Table 3.19. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.20. Body parameters

Parameter	Type	Description
body	ControllerConfig schema	

Table 3.21. HTTP responses

HTTP code	Reponse body
200 - OK	ControllerConfig schema
201 - Created	ControllerConfig schema
401 - Unauthorized	Empty

CHAPTER 4. CONTROLPLANEMACHINESET [MACHINE.OPENSIFT.IO/V1]

Description

ControlPlaneMachineSet ensures that a specified number of control plane machine replicas are running at any given time. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

4.1. SPECIFICATION

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

Property	Type	Description
status	object	ControlPlaneMachineSetStatus represents the status of the ControlPlaneMachineSet CRD.

4.1.1. .spec

Description

ControlPlaneMachineSet represents the configuration of the ControlPlaneMachineSet.

Type

object

Required

- **replicas**
- **selector**
- **template**

Property	Type	Description
replicas	integer	Replicas defines how many Control Plane Machines should be created by this ControlPlaneMachineSet. This field is immutable and cannot be changed after cluster installation. The ControlPlaneMachineSet only operates with 3 or 5 node control planes, 3 and 5 are the only valid values for this field.
selector	object	Label selector for Machines. Existing Machines selected by this selector will be the ones affected by this ControlPlaneMachineSet. It must match the template's labels. This field is considered immutable after creation of the resource.

Property	Type	Description
state	string	State defines whether the ControlPlaneMachineSet is Active or Inactive. When Inactive, the ControlPlaneMachineSet will not take any action on the state of the Machines within the cluster. When Active, the ControlPlaneMachineSet will reconcile the Machines and will update the Machines as necessary. Once Active, a ControlPlaneMachineSet cannot be made Inactive. To prevent further action please remove the ControlPlaneMachineSet.
strategy	object	Strategy defines how the ControlPlaneMachineSet will update Machines when it detects a change to the ProviderSpec.
template	object	Template describes the Control Plane Machines that will be created by this ControlPlaneMachineSet.

4.1.2. .spec.selector

Description

Label selector for Machines. Existing Machines selected by this selector will be the ones affected by this ControlPlaneMachineSet. It must match the template's labels. This field is considered immutable after creation of the resource.

Type

object

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

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

4.1.3. .spec.selector.matchExpressions

Description

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

Type

array

4.1.4. .spec.selector.matchExpressions[]

Description

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

Type

object

Required

- **key**
- **operator**

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

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

4.1.5. .spec.strategy

Description

Strategy defines how the ControlPlaneMachineSet will update Machines when it detects a change to the ProviderSpec.

Type

object

Property	Type	Description
type	string	Type defines the type of update strategy that should be used when updating Machines owned by the ControlPlaneMachineSet. Valid values are "RollingUpdate" and "OnDelete". The current default value is "RollingUpdate".

4.1.6. .spec.template

Description

Template describes the Control Plane Machines that will be created by this ControlPlaneMachineSet.

Type

object

Required

- **machineType**

Property	Type	Description
----------	------	-------------

Property	Type	Description
machineType	string	MachineType determines the type of Machines that should be managed by the ControlPlaneMachineSet. Currently, the only valid value is machines_v1beta1_machine_openshift_io.
machines_v1beta1_machine_openshift_io	object	OpenShiftMachineV1Beta1Machine defines the template for creating Machines from the v1beta1.machine.openshift.io API group.

4.1.7. .spec.template.machines_v1beta1_machine_openshift_io

Description

OpenShiftMachineV1Beta1Machine defines the template for creating Machines from the v1beta1.machine.openshift.io API group.

Type

object

Required

- **metadata**
- **spec**

Property	Type	Description
failureDomains	object	FailureDomains is the list of failure domains (sometimes called availability zones) in which the ControlPlaneMachineSet should balance the Control Plane Machines. This will be merged into the ProviderSpec given in the template. This field is optional on platforms that do not require placement information.

Property	Type	Description
metadata	object	ObjectMeta is the standard object metadata More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata Labels are required to match the ControlPlaneMachineSet selector.
spec	object	Spec contains the desired configuration of the Control Plane Machines. The ProviderSpec within contains platform specific details for creating the Control Plane Machines. The ProviderSe should be complete apart from the platform specific failure domain field. This will be overridden when the Machines are created based on the FailureDomains field.

4.1.8. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains

Description

FailureDomains is the list of failure domains (sometimes called availability zones) in which the ControlPlaneMachineSet should balance the Control Plane Machines. This will be merged into the ProviderSpec given in the template. This field is optional on platforms that do not require placement information.

Type

object

Required

- **platform**

Property	Type	Description
aws	array	AWS configures failure domain information for the AWS platform.
aws[]	object	AWSFailureDomain configures failure domain information for the AWS platform.

Property	Type	Description
azure	array	Azure configures failure domain information for the Azure platform.
azure[]	object	AzureFailureDomain configures failure domain information for the Azure platform.
gcp	array	GCP configures failure domain information for the GCP platform.
gcp[]	object	GCPFailureDomain configures failure domain information for the GCP platform
nutanix	array	nutanix configures failure domain information for the Nutanix platform.
nutanix[]	object	NutanixFailureDomainReference refers to the failure domain of the Nutanix platform.
openstack	array	OpenStack configures failure domain information for the OpenStack platform.
openstack[]	object	OpenStackFailureDomain configures failure domain information for the OpenStack platform.
platform	string	Platform identifies the platform for which the FailureDomain represents. Currently supported values are AWS, Azure, GCP, OpenStack, VSphere and Nutanix.
vsphere	array	vsphere configures failure domain information for the VSphere platform.
vsphere[]	object	VSphereFailureDomain configures failure domain information for the vSphere platform

4.1.9. .spec.template.machines_v1beta1_machine_opensift_io.failureDomains.aws

Description

AWS configures failure domain information for the AWS platform.

Type

array

4.1.10. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[]**Description**

AWSFailureDomain configures failure domain information for the AWS platform.

Type

object

Property	Type	Description
placement	object	Placement configures the placement information for this instance.
subnet	object	Subnet is a reference to the subnet to use for this instance.

4.1.11. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[].placement**Description**

Placement configures the placement information for this instance.

Type

object

Required

- **availabilityZone**

Property	Type	Description
availabilityZone	string	AvailabilityZone is the availability zone of the instance.

4.1.12. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[].placement.subnet**Description**

Subnet is a reference to the subnet to use for this instance.

Type

object

Required

- **type**

Property	Type	Description
arn	string	ARN of resource.
filters	array	Filters is a set of filters used to identify a resource.
filters[]	object	AWSResourceFilter is a filter used to identify an AWS resource
id	string	ID of resource.
type	string	Type determines how the reference will fetch the AWS resource.

4.1.13. `.spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[].si`

Description

Filters is a set of filters used to identify a resource.

Type

array

4.1.14. `.spec.template.machines_v1beta1_machine_openshift_io.failureDomains.aws[].si`

Description

AWSResourceFilter is a filter used to identify an AWS resource

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the filter. Filter names are case-sensitive.
values	array (string)	Values includes one or more filter values. Filter values are case-sensitive.

4.1.15. `.spec.template.machines_v1beta1_machine_openshift_io.failureDomains.azure`

Description

Azure configures failure domain information for the Azure platform.

Type**array****4.1.16. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.azure[]****Description**

AzureFailureDomain configures failure domain information for the Azure platform.

Type**object****Required**

- **zone**

Property	Type	Description
subnet	string	subnet is the name of the network subnet in which the VM will be created. When omitted, the subnet value from the machine providerSpec template will be used.
zone	string	Availability Zone for the virtual machine. If nil, the virtual machine should be deployed to no zone.

4.1.17. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.gcp**Description**

GCP configures failure domain information for the GCP platform.

Type**array****4.1.18. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.gcp[]****Description**

GCPFailureDomain configures failure domain information for the GCP platform

Type**object****Required**

- **zone**

Property	Type	Description
----------	------	-------------

Property	Type	Description
zone	string	Zone is the zone in which the GCP machine provider will create the VM.

4.1.19. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.nutanix

Description

nutanix configures failure domain information for the Nutanix platform.

Type

array

4.1.20. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.nutanix

Description

NutanixFailureDomainReference refers to the failure domain of the Nutanix platform.

Type

object

Required

- **name**

Property	Type	Description
name	string	name of the failure domain in which the nutanix machine provider will create the VM. Failure domains are defined in a cluster's config.openshift.io/Infrastructure resource.

4.1.21. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.openstack

Description

OpenStack configures failure domain information for the OpenStack platform.

Type

array

4.1.22. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.openstack

Description

OpenStackFailureDomain configures failure domain information for the OpenStack platform.

Type

object

Property	Type	Description
availabilityZone	string	availabilityZone is the nova availability zone in which the OpenStack machine provider will create the VM. If not specified, the VM will be created in the default availability zone specified in the nova configuration. Availability zone names must NOT contain : since it is used by admin users to specify hosts where instances are launched in server creation. Also, it must not contain spaces otherwise it will lead to node that belongs to this availability zone register failure, see kubernetes/cloud-provider-openstack#1379 for further information. The maximum length of availability zone name is 63 as per labels limits.
rootVolume	object	rootVolume contains settings that will be used by the OpenStack machine provider to create the root volume attached to the VM. If not specified, no root volume will be created.

4.1.23. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.openst

Description

rootVolume contains settings that will be used by the OpenStack machine provider to create the root volume attached to the VM. If not specified, no root volume will be created.

Type

object

Required

- **volumeType**

Property	Type	Description
----------	------	-------------

Property	Type	Description
availabilityZone	string	availabilityZone specifies the Cinder availability zone where the root volume will be created. If not specified, the root volume will be created in the availability zone specified by the volume type in the cinder configuration. If the volume type (configured in the OpenStack cluster) does not specify an availability zone, the root volume will be created in the default availability zone specified in the cinder configuration. See https://docs.openstack.org/cinder/latest/admin/availability-zone-type.html for more details. If the OpenStack cluster is deployed with the cross_az_attach configuration option set to false, the root volume will have to be in the same availability zone as the VM (defined by OpenStackFailureDomain.AvailabilityZone). Availability zone names must NOT contain spaces otherwise it will lead to volume that belongs to this availability zone register failure, see kubernetes/cloud-provider-openstack#1379 for further information. The maximum length of availability zone name is 63 as per labels limits.
volumeType	string	volumeType specifies the type of the root volume that will be provisioned. The maximum length of a volume type name is 255 characters, as per the OpenStack limit.

4.1.24. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.vsphere

Description

vsphere configures failure domain information for the VSphere platform.

Type

array

4.1.25. .spec.template.machines_v1beta1_machine_openshift_io.failureDomains.vsphere

Description

VSphereFailureDomain configures failure domain information for the vSphere platform

Type

object

Required

- **name**

Property	Type	Description
name	string	name of the failure domain in which the vSphere machine provider will create the VM. Failure domains are defined in a cluster's config.openshift.io/Infrastructure resource. When balancing machines across failure domains, the control plane machine set will inject configuration from the Infrastructure resource into the machine providerSpec to allocate the machine to a failure domain.

4.1.26. .spec.template.machines_v1beta1_machine_openshift_io.metadata**Description**

ObjectMeta is the standard object metadata More info:

<https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

Labels are required to match the ControlPlaneMachineSet selector.

Type

object

Required

- **labels**

Property	Type	Description
----------	------	-------------

Property	Type	Description
annotations	object (string)	Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. More info: http://kubernetes.io/docs/user-guide/annotations
labels	object (string)	Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services. More info: http://kubernetes.io/docs/user-guide/labels . This field must contain both the 'machine.openshift.io/cluster-api-machine-role' and 'machine.openshift.io/cluster-api-machine-type' labels, both with a value of 'master'. It must also contain a label with the key 'machine.openshift.io/cluster-api-cluster'.

4.1.27. .spec.template.machines_v1beta1_machine_openshift_io.spec

Description

Spec contains the desired configuration of the Control Plane Machines. The ProviderSpec within contains platform specific details for creating the Control Plane Machines. The ProviderSe should be complete apart from the platform specific failure domain field. This will be overridden when the Machines are created based on the FailureDomains field.

Type

object

Property	Type	Description
lifecycleHooks	object	LifecycleHooks allow users to pause operations on the machine at certain predefined points within the machine lifecycle.

Property	Type	Description
metadata	object	ObjectMeta will autopopulate the Node created. Use this to indicate what labels, annotations, name prefix, etc., should be used when creating the Node.
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.
providerSpec	object	ProviderSpec details Provider-specific configuration to use during node creation.

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

4.1.28. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks

Description

LifecycleHooks allow users to pause operations on the machine at certain predefined points within the machine lifecycle.

Type

object

Property	Type	Description
preDrain	array	PreDrain hooks prevent the machine from being drained. This also blocks further lifecycle events, such as termination.
preDrain[]	object	LifecycleHook represents a single instance of a lifecycle hook
preTerminate	array	PreTerminate hooks prevent the machine from being terminated. PreTerminate hooks be actioned after the Machine has been drained.
preTerminate[]	object	LifecycleHook represents a single instance of a lifecycle hook

4.1.29. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks.pr

Description

PreDrain hooks prevent the machine from being drained. This also blocks further lifecycle events, such as termination.

Type

array

4.1.30. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks.pr

Description

LifecycleHook represents a single instance of a lifecycle hook

Type

object

Required

- **name**
- **owner**

Property	Type	Description
name	string	Name defines a unique name for the lifecycle hook. The name should be unique and descriptive, ideally 1-3 words, in CamelCase or it may be namespaced, eg. foo.example.com/CamelCase. Names must be unique and should only be managed by a single entity.
owner	string	Owner defines the owner of the lifecycle hook. This should be descriptive enough so that users can identify who/what is responsible for blocking the lifecycle. This could be the name of a controller (e.g. clusteroperator/etcd) or an administrator managing the hook.

4.1.31. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks.pr

Description

PreTerminate hooks prevent the machine from being terminated. PreTerminate hooks be actioned after the Machine has been drained.

Type

array

4.1.32. .spec.template.machines_v1beta1_machine_openshift_io.spec.lifecycleHooks.pr

Description

LifecycleHook represents a single instance of a lifecycle hook

Type

object

Required

- **name**
- **owner**

Property	Type	Description
name	string	Name defines a unique name for the lifecycle hook. The name should be unique and descriptive, ideally 1-3 words, in CamelCase or it may be namespaced, eg. foo.example.com/CamelCase. Names must be unique and should only be managed by a single entity.
owner	string	Owner defines the owner of the lifecycle hook. This should be descriptive enough so that users can identify who/what is responsible for blocking the lifecycle. This could be the name of a controller (e.g. clusteroperator/etcd) or an administrator managing the hook.

4.1.33. .spec.template.machines_v1beta1_machine_openshift_io.spec.metadata

Description

ObjectMeta will autopopulate the Node created. Use this to indicate what labels, annotations, name prefix, etc., should be used when creating the Node.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
annotations	object (string)	Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. More info: http://kubernetes.io/docs/user-guide/annotations
generateName	string	GenerateName is an optional prefix, used by the server, to generate a unique name ONLY IF the Name field has not been provided. If this field is used, the name returned to the client will be different than the name passed. This value will also be combined with a unique suffix. The provided value has the same validation rules as the Name field, and may be truncated by the length of the suffix required to make the value unique on the server. If this field is specified and the generated name exists, the server will NOT return a 409 - instead, it will either return 201 Created or 500 with Reason ServerTimeout indicating a unique name could not be found in the time allotted, and the client should retry (optionally after the time indicated in the Retry-After header). Applied only if Name is not specified. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#idempotency
labels	object (string)	Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services. More info: http://kubernetes.io/docs/user-guide/labels

Property	Type	Description
name	string	Name must be unique within a namespace. Is required when creating resources, although some resources may allow a client to request the generation of an appropriate name automatically. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names
namespace	string	Namespace defines the space within each name must be unique. An empty namespace is equivalent to the "default" namespace, but "default" is the canonical representation. Not all objects are required to be scoped to a namespace - the value of this field for those objects will be empty. Must be a DNS_LABEL. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/namespaces
ownerReferences	array	List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.
ownerReferences[]	object	OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

4.1.34. .spec.template.machines_v1beta1_machine_openshift_io.spec.metadata.ownerF

Description

List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to

this controller, with the controller field set to true. There cannot be more than one managing controller.

Type

array

4.1.35. .spec.template.machines_v1beta1_machine_openshift_io.spec.metadata.ownerReference

Description

OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

Type

object

Required

- **apiVersion**
- **kind**
- **name**
- **uid**

Property	Type	Description
apiVersion	string	API version of the referent.
blockOwnerDeletion	boolean	If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is removed. See https://kubernetes.io/docs/concepts/architecture/garbage-collection/#foreground-deletion for how the garbage collector interacts with this field and enforces the foreground deletion. Defaults to false. To set this field, a user needs "delete" permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.
controller	boolean	If true, this reference points to the managing controller.

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

4.1.36. .spec.template.machines_v1beta1_machine_openshift_io.spec.providerSpec

Description

ProviderSpec details Provider-specific configuration to use during node creation.

Type

object

Property	Type	Description
value	..	Value is an inlined, serialized representation of the resource configuration. It is recommended that providers maintain their own versioned API types that should be serialized/deserialized from this field, akin to component config.

4.1.37. .spec.template.machines_v1beta1_machine_openshift_io.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

4.1.38. .spec.template.machines_v1beta1_machine_openshift_io.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.

4.1.39. .status**Description**

ControlPlaneMachineSetStatus represents the status of the ControlPlaneMachineSet CRD.

Type

object

Property	Type	Description
conditions	array	Conditions represents the observations of the ControlPlaneMachineSet's current state. Known .status.conditions.type are: Available, Degraded and Progressing.

Property	Type	Description
conditions[]	object	<p>Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions" // other fields }</p>
observedGeneration	integer	ObservedGeneration is the most recent generation observed for this ControlPlaneMachineSet. It corresponds to the ControlPlaneMachineSets's generation, which is updated on mutation by the API Server.
readyReplicas	integer	ReadyReplicas is the number of Control Plane Machines created by the ControlPlaneMachineSet controller which are ready. Note that this value may be higher than the desired number of replicas while rolling updates are in-progress.
replicas	integer	Replicas is the number of Control Plane Machines created by the ControlPlaneMachineSet controller. Note that during update operations this value may differ from the desired replica count.

Property	Type	Description
unavailableReplicas	integer	UnavailableReplicas is the number of Control Plane Machines that are still required before the ControlPlaneMachineSet reaches the desired available capacity. When this value is non-zero, the number of ReadyReplicas is less than the desired Replicas.
updatedReplicas	integer	UpdatedReplicas is the number of non-terminated Control Plane Machines created by the ControlPlaneMachineSet controller that have the desired provider spec and are ready. This value is set to 0 when a change is detected to the desired spec. When the update strategy is RollingUpdate, this will also coincide with starting the process of updating the Machines. When the update strategy is OnDelete, this value will remain at 0 until a user deletes an existing replica and its replacement has become ready.

4.1.40. .status.conditions

Description

Conditions represents the observations of the ControlPlaneMachineSet's current state. Known `.status.conditions.type` are: Available, Degraded and Progressing.

Type

array

4.1.41. .status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path `.status.conditions`. For example, type `FooStatus`

```
struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions" // other fields }
```

Type

object

Required

- **lastTransitionTime**
- **message**
- **reason**
- **status**
- **type**

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

Property	Type	Description
type	string	type of condition in CamelCase or in foo.example.com/CamelCase. -- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)? (qualifiedNameFmt)

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machine.openshift.io/v1/controlplanemachinesets**
 - **GET**: list objects of kind ControlPlaneMachineSet
- **/apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachinesets**
 - **DELETE**: delete collection of ControlPlaneMachineSet
 - **GET**: list objects of kind ControlPlaneMachineSet
 - **POST**: create a ControlPlaneMachineSet
- **/apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachinesets/{name}**
 - **DELETE**: delete a ControlPlaneMachineSet
 - **GET**: read the specified ControlPlaneMachineSet
 - **PATCH**: partially update the specified ControlPlaneMachineSet
 - **PUT**: replace the specified ControlPlaneMachineSet
- **/apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachinesets/{name}/scale**
 - **GET**: read scale of the specified ControlPlaneMachineSet
 - **PATCH**: partially update scale of the specified ControlPlaneMachineSet
 - **PUT**: replace scale of the specified ControlPlaneMachineSet
- **/apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachinesets/{name}/status**

- **GET**: read status of the specified ControlPlaneMachineSet
- **PATCH**: partially update status of the specified ControlPlaneMachineSet
- **PUT**: replace status of the specified ControlPlaneMachineSet

4.2.1. /apis/machine.openshift.io/v1/controlplanemachinesets

HTTP method

GET

Description

list objects of kind ControlPlaneMachineSet

Table 4.1. HTTP responses

HTTP code	Reponse body
200 - OK	ControlPlaneMachineSetList schema
401 - Unauthorized	Empty

4.2.2. /apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachine

HTTP method

DELETE

Description

delete collection of ControlPlaneMachineSet

Table 4.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind ControlPlaneMachineSet

Table 4.3. HTTP responses

HTTP code	Reponse body
200 - OK	ControlPlaneMachineSetList schema
401 - Unauthorized	Empty

HTTP code	Response body
-----------	---------------

HTTP method

POST

Description

create a ControlPlaneMachineSet

Table 4.4. Query parameters

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

Table 4.5. Body parameters

Parameter	Type	Description
body	ControlPlaneMachineSet schema	

Table 4.6. HTTP responses

HTTP code	Reponse body
200 - OK	ControlPlaneMachineSet schema
201 - Created	ControlPlaneMachineSet schema
202 - Accepted	ControlPlaneMachineSet schema
401 - Unauthorized	Empty

4.2.3. /apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachine

Table 4.7. Global path parameters

Parameter	Type	Description
name	string	name of the ControlPlaneMachineSet

HTTP method

DELETE

Description

delete a ControlPlaneMachineSet

Table 4.8. Query parameters

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

Table 4.9. HTTP responses

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

HTTP method

GET

Description

read the specified ControlPlaneMachineSet

Table 4.10. HTTP responses

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

HTTP method**PATCH****Description**

partially update the specified ControlPlaneMachineSet

Table 4.11. Query parameters

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

Table 4.12. HTTP responses

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

HTTP method**PUT****Description**

replace the specified ControlPlaneMachineSet

Table 4.13. Query parameters

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

Table 4.14. Body parameters

Parameter	Type	Description
body	ControlPlaneMachineSet schema	

Table 4.15. HTTP responses

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

4.2.4. /apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachine

Table 4.16. Global path parameters

Parameter	Type	Description
name	string	name of the ControlPlaneMachineSet

HTTP method

GET

Description

read scale of the specified ControlPlaneMachineSet

Table 4.17. HTTP responses

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

HTTP method

PATCH

Description

partially update scale of the specified ControlPlaneMachineSet

Table 4.18. Query parameters

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

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

Table 4.19. HTTP responses

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

HTTP method**PUT****Description**

replace scale of the specified ControlPlaneMachineSet

Table 4.20. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 4.21. Body parameters

Parameter	Type	Description
body	Scale schema	

Table 4.22. HTTP responses

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

4.2.5. /apis/machine.openshift.io/v1/namespaces/{namespace}/controlplanemachine

Table 4.23. Global path parameters

Parameter	Type	Description
name	string	name of the ControlPlaneMachineSet

HTTP method

GET

Description

read status of the specified ControlPlaneMachineSet

Table 4.24. HTTP responses

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

HTTP method**PATCH****Description**

partially update status of the specified ControlPlaneMachineSet

Table 4.25. Query parameters

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

Table 4.26. HTTP responses

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

HTTP method**PUT****Description**

replace status of the specified ControlPlaneMachineSet

Table 4.27. Query parameters

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

Table 4.28. Body parameters

Parameter	Type	Description
body	ControlPlaneMachineSet schema	

Table 4.29. HTTP responses

HTTP code	Reponse body
200 - OK	ControlPlaneMachineSet schema
201 - Created	ControlPlaneMachineSet schema
401 - Unauthorized	Empty

CHAPTER 5. KUBELETCONFIG [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

KubeletConfig describes a customized Kubelet configuration. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

5.1. SPECIFICATION

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

Property	Type	Description
status	object	KubeletConfigStatus defines the observed state of a KubeletConfig

5.1.1. .spec

Description

KubeletConfigSpec defines the desired state of KubeletConfig

Type

object

Property	Type	Description
autoSizingReserved	boolean	
kubeletConfig	object	kubeletConfig fields are defined in kubernetes upstream. Please refer to the types defined in the version/commit used by OpenShift of the upstream kubernetes. It's important to note that, since the fields of the kubelet configuration are directly fetched from upstream the validation of those values is handled directly by the kubelet. Please refer to the upstream version of the relevant kubernetes for the valid values of these fields. Invalid values of the kubelet configuration fields may render cluster nodes unusable.
logLevel	integer	
machineConfigPoolSelector	object	MachineConfigPoolSelector selects which pools the KubeletConfig should apply to. A nil selector will result in no pools being selected.

Property	Type	Description
tlsSecurityProfile	object	If unset, the default is based on the <code>apiservers.config.openshift.io/cluster</code> resource. Note that only Old and Intermediate profiles are currently supported, and the maximum available <code>minTLSVersion</code> is <code>VersionTLS12</code> .

5.1.2. `.spec.machineConfigPoolSelector`

Description

`MachineConfigPoolSelector` selects which pools the `KubeletConfig` should apply to. A nil selector will result in no pools being selected.

Type

object

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

5.1.3. `.spec.machineConfigPoolSelector.matchExpressions`

Description

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

Type

array

5.1.4. `.spec.machineConfigPoolSelector.matchExpressions[]`

Description

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

Type

object

Required

- **key**
- **operator**

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

5.1.5. .spec.tlsSecurityProfile**Description**

If unset, the default is based on the apiservers.config.openshift.io/cluster resource. Note that only Old and Intermediate profiles are currently supported, and the maximum available minTLSVersion is VersionTLS12.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
custom	''	<p>custom is a user-defined TLS security profile. Be extremely careful using a custom profile as invalid configurations can be catastrophic. An example custom profile looks like this: ciphers: - ECDHE-ECDSA-CHACHA20-POLY1305 - ECDHE-RSA-CHACHA20-POLY1305 - ECDHE-RSA-AES128-GCM-SHA256 - ECDHE-ECDSA-AES128-GCM-SHA256 minTLSVersion: VersionTLS11</p>
intermediate	''	<p>intermediate is a TLS security profile based on: https://wiki.mozilla.org/Security/Server_Side_TLS#Intermediate_compatibility_28recommended.29 and looks like this (yaml): ciphers: - TLS_AES_128_GCM_SHA256 - TLS_AES_256_GCM_SHA384 - TLS_CHACHA20_POLY1305_SHA256 - ECDHE-ECDSA-AES128-GCM-SHA256 - ECDHE-RSA-AES128-GCM-SHA256 - ECDHE-ECDSA-AES256-GCM-SHA384 - ECDHE-RSA-AES256-GCM-SHA384 - ECDHE-ECDSA-CHACHA20-POLY1305 - ECDHE-RSA-CHACHA20-POLY1305 - DHE-RSA-AES128-GCM-SHA256 - DHE-RSA-AES256-GCM-SHA384 minTLSVersion: VersionTLS12</p>
modern	''	<p>modern is a TLS security profile based on: https://wiki.mozilla.org/Security/Server_Side_TLS#Modern_compatibility and looks like this (yaml): ciphers: - TLS_AES_128_GCM_SHA256 - TLS_AES_256_GCM_SHA384 - TLS_CHACHA20_POLY1305_SHA256 minTLSVersion: VersionTLS13</p>

Property	Type	Description
old	..	<p>old is a TLS security profile based on:</p> <p>https://wiki.mozilla.org/Security/Server_Side_TLS#Old_backward_compatibility and looks like this (yaml):</p> <pre> ciphers: - TLS_AES_128_GCM_SHA256 - TLS_AES_256_GCM_SHA384 - TLS_CHACHA20_POLY1305_SHA256 - ECDHE-ECDSA-AES128-GCM-SHA256 - ECDHE-RSA-GCM-SHA256 - ECDHE-AES128-GCM-SHA256 - ECDHE-ECDSA-AES256-GCM-SHA384 - ECDHE-RSA-AES256-GCM-SHA384 - ECDHE-ECDSA-CHACHA20-POLY1305 - ECDHE-RSA-CHACHA20-POLY1305 - DHE-RSA-AES128-GCM-SHA256 - DHE-RSA-AES256-GCM-SHA384 - DHE-RSA-CHACHA20-POLY1305 - ECDHE-ECDSA-AES128-SHA256 - ECDHE-RSA-AES128-SHA256 - ECDHE-ECDSA-AES128-SHA - ECDHE-RSA-AES128-SHA - ECDHE-ECDSA-AES256-SHA384 - ECDHE-RSA-AES256-SHA384 - ECDHE-ECDSA-AES256-SHA - ECDHE-RSA-AES256-SHA - DHE-RSA-AES128-SHA256 - DHE-RSA-AES256-SHA256 - AES128-GCM-SHA256 - AES256-GCM-SHA384 - AES128-SHA256 - AES256-SHA256 - AES128-SHA - AES256-SHA - DES-CBC3-SHA minTLSVersion: VersionTLS10 </pre>

Property	Type	Description
type	string	type is one of Old, Intermediate, Modern or Custom. Custom provides the ability to specify individual TLS security profile parameters. Old, Intermediate and Modern are TLS security profiles based on: https://wiki.mozilla.org/Security/Server_Side_TLS#Recommended_configurations The profiles are intent based, so they may change over time as new ciphers are developed and existing ciphers are found to be insecure. Depending on precisely which ciphers are available to a process, the list may be reduced. Note that the Modern profile is currently not supported because it is not yet well adopted by common software libraries.

5.1.6. .status

Description

KubeletConfigStatus defines the observed state of a KubeletConfig

Type

object

Property	Type	Description
conditions	array	conditions represents the latest available observations of current state.
conditions[]	object	KubeletConfigCondition defines the state of the KubeletConfig
observedGeneration	integer	observedGeneration represents the generation observed by the controller.

5.1.7. .status.conditions

Description

conditions represents the latest available observations of current state.

Type

array

5.1.8. .status.conditions[]

Description

KubeletConfigCondition defines the state of the KubeletConfig

Type

object

Property	Type	Description
lastTransitionTime	..	lastTransitionTime is the time of the last update to the current status object.
message	string	message provides additional information about the current condition. This is only to be consumed by humans.
reason	string	reason is the reason for the condition's last transition. Reasons are PascalCase
status	string	status of the condition, one of True, False, Unknown.
type	string	type specifies the state of the operator's reconciliation functionality.

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machineconfiguration.openshift.io/v1/kubeletconfigs**
 - **DELETE:** delete collection of KubeletConfig
 - **GET:** list objects of kind KubeletConfig
 - **POST:** create a KubeletConfig
- **/apis/machineconfiguration.openshift.io/v1/kubeletconfigs/{name}**
 - **DELETE:** delete a KubeletConfig
 - **GET:** read the specified KubeletConfig
 - **PATCH:** partially update the specified KubeletConfig
 - **PUT:** replace the specified KubeletConfig

- **/apis/machineconfiguration.openshift.io/v1/kubeletconfigs/{name}/status**
 - **GET**: read status of the specified KubeletConfig
 - **PATCH**: partially update status of the specified KubeletConfig
 - **PUT**: replace status of the specified KubeletConfig

5.2.1. /apis/machineconfiguration.openshift.io/v1/kubeletconfigs

HTTP method

DELETE

Description

delete collection of KubeletConfig

Table 5.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind KubeletConfig

Table 5.2. HTTP responses

HTTP code	Reponse body
200 - OK	KubeletConfigList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a KubeletConfig

Table 5.3. Query parameters

Parameter	Type	Description
-----------	------	-------------

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

Table 5.4. Body parameters

Parameter	Type	Description
body	KubeletConfig schema	

Table 5.5. HTTP responses

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

5.2.2. /apis/machineconfiguration.openshift.io/v1/kubeletconfigs/{name}

Table 5.6. Global path parameters

Parameter	Type	Description
name	string	name of the KubeletConfig

HTTP method

DELETE

Description

delete a KubeletConfig

Table 5.7. Query parameters

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

Table 5.8. HTTP responses

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

HTTP method

GET

Description

read the specified KubeletConfig

Table 5.9. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified KubeletConfig

Table 5.10. Query parameters

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

Table 5.11. HTTP responses

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

HTTP method**PUT****Description**

replace the specified KubeletConfig

Table 5.12. Query parameters

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

Table 5.13. Body parameters

Parameter	Type	Description
body	KubeletConfig schema	

Table 5.14. HTTP responses

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

5.2.3. /apis/machineconfiguration.openshift.io/v1/kubeletconfigs/{name}/status

Table 5.15. Global path parameters

Parameter	Type	Description
name	string	name of the KubeletConfig

HTTP method

GET

Description

read status of the specified KubeletConfig

Table 5.16. HTTP responses

HTTP code	Reponse body
200 - OK	KubeletConfig schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified KubeletConfig

Table 5.17. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.18. HTTP responses

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

HTTP method**PUT****Description**

replace status of the specified KubeletConfig

Table 5.19. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.20. Body parameters

Parameter	Type	Description
body	KubeletConfig schema	

Table 5.21. HTTP responses

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

CHAPTER 6. MACHINECONFIG

[MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

MachineConfig defines the configuration for a machine Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

6.1. SPECIFICATION

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

6.1.1. .spec

Description

MachineConfigSpec is the spec for MachineConfig

Type
object

Property	Type	Description
baseOSExtensionsContainerImage	string	BaseOSExtensionsContainerImage specifies the remote location that will be used to fetch the extensions container matching a new-format OS image
config	``	Config is a Ignition Config object.
extensions	array (string)	extensions contains a list of additional features that can be enabled on host
fips	boolean	fips controls FIPS mode
kernelArguments	``	kernelArguments contains a list of kernel arguments to be added
kernelType	string	kernelType contains which kernel we want to be running like default (traditional), realtime, 64k-pages (aarch64 only).
osImageURL	string	OSImageURL specifies the remote location that will be used to fetch the OS.

6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machineconfiguration.openshift.io/v1/machineconfigs**
 - **DELETE**: delete collection of MachineConfig
 - **GET**: list objects of kind MachineConfig
 - **POST**: create a MachineConfig
- **/apis/machineconfiguration.openshift.io/v1/machineconfigs/{name}**
 - **DELETE**: delete a MachineConfig
 - **GET**: read the specified MachineConfig
 - **PATCH**: partially update the specified MachineConfig
 - **PUT**: replace the specified MachineConfig

6.2.1. /apis/machineconfiguration.openshift.io/v1/machineconfigs

HTTP method

DELETE

Description

delete collection of MachineConfig

Table 6.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind MachineConfig

Table 6.2. HTTP responses

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

HTTP method

POST

Description

create a MachineConfig

Table 6.3. Query parameters

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

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: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 6.4. Body parameters

Parameter	Type	Description
body	MachineConfig schema	

Table 6.5. HTTP responses

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

6.2.2. /apis/machineconfiguration.openshift.io/v1/machineconfigs/{name}

Table 6.6. Global path parameters

Parameter	Type	Description
name	string	name of the MachineConfig

HTTP method**DELETE****Description**

delete a MachineConfig

Table 6.7. Query parameters

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

Table 6.8. HTTP responses

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

HTTP method**GET****Description**

read the specified MachineConfig

Table 6.9. HTTP responses

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

HTTP method**PATCH****Description**

partially update the specified MachineConfig

Table 6.10. Query parameters

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

Table 6.11. HTTP responses

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

HTTP method**PUT****Description**

replace the specified MachineConfig

Table 6.12. Query parameters

Parameter	Type	Description
-----------	------	-------------

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

Table 6.13. Body parameters

Parameter	Type	Description
body	MachineConfig schema	

Table 6.14. HTTP responses

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

CHAPTER 7. MACHINECONFIGPOOL [MACHINECONFIGURATION.OPENSIFT.IO/V1]

Description

MachineConfigPool describes a pool of MachineConfigs. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

7.1. SPECIFICATION

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

Property	Type	Description
status	object	MachineConfigPoolStatus is the status for MachineConfigPool resource.

7.1.1. .spec

Description

MachineConfigPoolSpec is the spec for MachineConfigPool resource.

Type

object

Property	Type	Description
configuration	object	The targeted MachineConfig object for the machine config pool.
machineConfigSelector	object	machineConfigSelector specifies a label selector for MachineConfigs. Refer https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/ on how label and selectors work.
maxUnavailable	integer-or-string	maxUnavailable defines either an integer number or percentage of nodes in the pool that can go Unavailable during an update. This includes nodes Unavailable for any reason, including user initiated cordons, failing nodes, etc. The default value is 1. A value larger than 1 will mean multiple nodes going unavailable during the update, which may affect your workload stress on the remaining nodes. You cannot set this value to 0 to stop updates (it will default back to 1); to stop updates, use the 'paused' property instead. Drain will respect Pod Disruption Budgets (PDBs) such as etcd quorum guards, even if maxUnavailable is greater than one.

Property	Type	Description
nodeSelector	object	nodeSelector specifies a label selector for Machines
paused	boolean	paused specifies whether or not changes to this machine config pool should be stopped. This includes generating new desiredMachineConfig and update of machines.

7.1.2. .spec.configuration

Description

The targeted MachineConfig object for the machine config pool.

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 <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.

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
source	array	source is the list of MachineConfig objects that were used to generate the single MachineConfig object specified in content .

Property	Type	Description
source[]	object	<p>ObjectReference contains enough information to let you inspect or modify the referred object. --- New uses of this type are discouraged because of difficulty describing its usage when embedded in APIs. 1. Ignored fields. It includes many fields which are not generally honored. For instance, ResourceVersion and FieldPath are both very rarely valid in actual usage. 2. Invalid usage help. It is impossible to add specific help for individual usage. In most embedded usages, there are particular restrictions like, "must refer only to types A and B" or "UID not honored" or "name must be restricted". Those cannot be well described when embedded. 3. Inconsistent validation. Because the usages are different, the validation rules are different by usage, which makes it hard for users to predict what will happen. 4. The fields are both imprecise and overly precise. Kind is not a precise mapping to a URL. This can produce ambiguity during interpretation and require a REST mapping. In most cases, the dependency is on the group,resource tuple and the version of the actual struct is irrelevant. 5. We cannot easily change it. Because this type is embedded in many locations, updates to this type will affect numerous schemas. Don't make new APIs embed an underspecified API type they do not control. Instead of using this type, create a locally provided and used type that is well-focused on your reference. For example, ServiceReferences for admission registration: https://github.com/kubernetes/api/blob/release-1.17/admissionregistration/v1/types.go#L533 .</p>

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

7.1.3. .spec.configuration.source

Description

source is the list of MachineConfig objects that were used to generate the single MachineConfig object specified in **content**.

Type

array

7.1.4. .spec.configuration.source[]

Description

ObjectReference contains enough information to let you inspect or modify the referred object. --- New uses of this type are discouraged because of difficulty describing its usage when embedded in APIs. 1. Ignored fields. It includes many fields which are not generally honored. For instance, ResourceVersion and FieldPath are both very rarely valid in actual usage. 2. Invalid usage help. It is impossible to add specific help for individual usage. In most embedded usages, there are particular restrictions like, "must refer only to types A and B" or "UID not honored" or "name must be restricted". Those cannot be well described when embedded. 3. Inconsistent validation. Because the usages are different, the validation rules are different by usage, which makes it hard for users to predict what will happen. 4. The fields are both imprecise and overly precise. Kind is not a precise mapping to a URL. This can produce ambiguity during interpretation and require a REST mapping. In most cases, the dependency is on the group,resource tuple and the version of the actual struct is irrelevant. 5. We cannot easily change it. Because this type is embedded in many locations, updates to this type will affect numerous schemas. Don't make new APIs embed an underspecified API type they do not control. Instead of using this type, create a locally provided and used type that is well-focused on your reference. For example, ServiceReferences for admission registration: <https://github.com/kubernetes/api/blob/release-1.17/admissionregistration/v1/types.go#L533> .

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 <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: 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

7.1.5. .spec.machineConfigSelector

Description

machineConfigSelector specifies a label selector for MachineConfigs. Refer <https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/> on how label and selectors work.

Type

object

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

7.1.6. .spec.machineConfigSelector.matchExpressions

Description

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

Type

array

7.1.7. .spec.machineConfigSelector.matchExpressions[]

Description

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

Type

object

Required

- **key**
- **operator**

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

7.1.8. .spec.nodeSelector

Description

nodeSelector specifies a label selector for Machines

Type

object

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

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

7.1.9. .spec.nodeSelector.matchExpressions

Description

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

Type

array

7.1.10. .spec.nodeSelector.matchExpressions[]

Description

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

Type

object

Required

- **key**
- **operator**

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

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

7.1.11. .status

Description

MachineConfigPoolStatus is the status for MachineConfigPool resource.

Type

object

Property	Type	Description
certExpirys	array	certExpirys keeps track of important certificate expiration data
certExpirys[]	object	ceryExpiry contains the bundle name and the expiry date
conditions	array	conditions represents the latest available observations of current state.
conditions[]	object	MachineConfigPoolCondition contains condition information for an MachineConfigPool.
configuration	object	configuration represents the current MachineConfig object for the machine config pool.
degradedMachineCount	integer	degradedMachineCount represents the total number of machines marked degraded (or unreconcilable). A node is marked degraded if applying a configuration failed..
machineCount	integer	machineCount represents the total number of machines in the machine config pool.

Property	Type	Description
observedGeneration	integer	observedGeneration represents the generation observed by the controller.
readyMachineCount	integer	readyMachineCount represents the total number of ready machines targeted by the pool.
unavailableMachineCount	integer	unavailableMachineCount represents the total number of unavailable (non-ready) machines targeted by the pool. A node is marked unavailable if it is in updating state or NodeReady condition is false.
updatedMachineCount	integer	updatedMachineCount represents the total number of machines targeted by the pool that have the CurrentMachineConfig as their config.

7.1.12. .status.certExpirys

Description

certExpirys keeps track of important certificate expiration data

Type

array

7.1.13. .status.certExpirys[]

Description

certExpiry contains the bundle name and the expiry date

Type

object

Required

- **bundle**
- **subject**

Property	Type	Description
----------	------	-------------

Property	Type	Description
bundle	string	bundle is the name of the bundle in which the subject certificate resides
expiry	string	expiry is the date after which the certificate will no longer be valid
subject	string	subject is the subject of the certificate

7.1.14. .status.conditions

Description

conditions represents the latest available observations of current state.

Type

array

7.1.15. .status.conditions[]

Description

MachineConfigPoolCondition contains condition information for an MachineConfigPool.

Type

object

Property	Type	Description
lastTransitionTime	string	lastTransitionTime is the timestamp corresponding to the last status change of this condition.
message	string	message is a human readable description of the details of the last transition, complementing reason.
reason	string	reason is a brief machine readable explanation for the condition's last transition.
status	string	status of the condition, one of ('True', 'False', 'Unknown').
type	string	type of the condition, currently ('Done', 'Updating', 'Failed').

7.1.16. .status.configuration

Description

configuration represents the current MachineConfig object for the machine config pool.

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 <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

Property	Type	Description
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
source	array	source is the list of MachineConfig objects that were used to generate the single MachineConfig object specified in content .

Property	Type	Description
source[]	object	<p>ObjectReference contains enough information to let you inspect or modify the referred object. --- New uses of this type are discouraged because of difficulty describing its usage when embedded in APIs. 1. Ignored fields. It includes many fields which are not generally honored. For instance, ResourceVersion and FieldPath are both very rarely valid in actual usage. 2. Invalid usage help. It is impossible to add specific help for individual usage. In most embedded usages, there are particular restrictions like, "must refer only to types A and B" or "UID not honored" or "name must be restricted". Those cannot be well described when embedded. 3. Inconsistent validation. Because the usages are different, the validation rules are different by usage, which makes it hard for users to predict what will happen. 4. The fields are both imprecise and overly precise. Kind is not a precise mapping to a URL. This can produce ambiguity during interpretation and require a REST mapping. In most cases, the dependency is on the group,resource tuple and the version of the actual struct is irrelevant. 5. We cannot easily change it. Because this type is embedded in many locations, updates to this type will affect numerous schemas. Don't make new APIs embed an underspecified API type they do not control. Instead of using this type, create a locally provided and used type that is well-focused on your reference. For example, ServiceReferences for admission registration: https://github.com/kubernetes/api/blob/release-1.17/admissionregistration/v1/types.go#L533 .</p>

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

7.1.17. .status.configuration.source

Description

source is the list of MachineConfig objects that were used to generate the single MachineConfig object specified in **content**.

Type

array

7.1.18. .status.configuration.source[]

Description

ObjectReference contains enough information to let you inspect or modify the referred object. --- New uses of this type are discouraged because of difficulty describing its usage when embedded in APIs. 1. Ignored fields. It includes many fields which are not generally honored. For instance, ResourceVersion and FieldPath are both very rarely valid in actual usage. 2. Invalid usage help. It is impossible to add specific help for individual usage. In most embedded usages, there are particular restrictions like, "must refer only to types A and B" or "UID not honored" or "name must be restricted". Those cannot be well described when embedded. 3. Inconsistent validation. Because the usages are different, the validation rules are different by usage, which makes it hard for users to predict what will happen. 4. The fields are both imprecise and overly precise. Kind is not a precise mapping to a URL. This can produce ambiguity during interpretation and require a REST mapping. In most cases, the dependency is on the group,resource tuple and the version of the actual struct is irrelevant. 5. We cannot easily change it. Because this type is embedded in many locations, updates to this type will affect numerous schemas. Don't make new APIs embed an underspecified API type they do not control. Instead of using this type, create a locally provided and used type that is well-focused on your reference. For example, ServiceReferences for admission registration: <https://github.com/kubernetes/api/blob/release-1.17/admissionregistration/v1/types.go#L533> .

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 <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: 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

7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machineconfiguration.openshift.io/v1/machineconfigpools**
 - **DELETE**: delete collection of MachineConfigPool
 - **GET**: list objects of kind MachineConfigPool
 - **POST**: create a MachineConfigPool
- **/apis/machineconfiguration.openshift.io/v1/machineconfigpools/{name}**
 - **DELETE**: delete a MachineConfigPool
 - **GET**: read the specified MachineConfigPool
 - **PATCH**: partially update the specified MachineConfigPool
 - **PUT**: replace the specified MachineConfigPool
- **/apis/machineconfiguration.openshift.io/v1/machineconfigpools/{name}/status**
 - **GET**: read status of the specified MachineConfigPool
 - **PATCH**: partially update status of the specified MachineConfigPool
 - **PUT**: replace status of the specified MachineConfigPool

7.2.1. /apis/machineconfiguration.openshift.io/v1/machineconfigpools

HTTP method

DELETE

Description

delete collection of MachineConfigPool

Table 7.1. HTTP responses

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

HTTP method**GET****Description**

list objects of kind MachineConfigPool

Table 7.2. HTTP responses

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

HTTP method**POST****Description**

create a MachineConfigPool

Table 7.3. Query parameters

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

Table 7.4. Body parameters

Parameter	Type	Description
body	MachineConfigPool schema	

Table 7.5. HTTP responses

HTTP code	Reponse body
200 - OK	MachineConfigPool schema
201 - Created	MachineConfigPool schema
202 - Accepted	MachineConfigPool schema
401 - Unauthorized	Empty

7.2.2. /apis/machineconfiguration.openshift.io/v1/machineconfigpools/{name}

Table 7.6. Global path parameters

Parameter	Type	Description
name	string	name of the MachineConfigPool

HTTP method

DELETE

Description

delete a MachineConfigPool

Table 7.7. Query parameters

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

Table 7.8. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema

HTTP code	Reponse body
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

read the specified MachineConfigPool

Table 7.9. HTTP responses

HTTP code	Reponse body
200 - OK	MachineConfigPool schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified MachineConfigPool

Table 7.10. Query parameters

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

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

Table 7.11. HTTP responses

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

HTTP method**PUT****Description**

replace the specified MachineConfigPool

Table 7.12. Query parameters

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

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: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 7.13. Body parameters

Parameter	Type	Description
body	MachineConfigPool schema	

Table 7.14. HTTP responses

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

7.2.3. /apis/machineconfiguration.openshift.io/v1/machineconfigpools/{name}/status

Table 7.15. Global path parameters

Parameter	Type	Description
name	string	name of the MachineConfigPool

HTTP method

GET**Description**

read status of the specified MachineConfigPool

Table 7.16. HTTP responses

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

HTTP method**PATCH****Description**

partially update status of the specified MachineConfigPool

Table 7.17. Query parameters

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

Table 7.18. HTTP responses

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

HTTP method**PUT****Description**

replace status of the specified MachineConfigPool

Table 7.19. Query parameters

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

Table 7.20. Body parameters

Parameter	Type	Description
body	MachineConfigPool schema	

Table 7.21. HTTP responses

HTTP code	Reponse body
200 - OK	MachineConfigPool schema
201 - Created	MachineConfigPool schema
401 - Unauthorized	Empty

CHAPTER 8. MACHINEHEALTHCHECK [MACHINE.OPENSIFT.IO/V1BETA1]

Description

MachineHealthCheck is the Schema for the machinehealthchecks API Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

8.1. SPECIFICATION

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

8.1.1. .spec

Description

Specification of machine health check policy

Type

object

Property	Type	Description
maxUnhealthy	integer-or-string	Any farther remediation is only allowed if at most "MaxUnhealthy" machines selected by "selector" are not healthy. Expects either a positive integer value or a percentage value. Percentage values must be positive whole numbers and are capped at 100%. Both 0 and 0% are valid and will block all remediation.
nodeStartupTimeout	string	Machines older than this duration without a node will be considered to have failed and will be remediated. To prevent Machines without Nodes from being removed, disable startup checks by setting this value explicitly to "0". Expects an unsigned duration string of decimal numbers each with optional fraction and a unit suffix, eg "300ms", "1.5h" or "2h45m". Valid time units are "ns", "us" (or "µs"), "ms", "s", "m", "h".
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 Machine API Operator.
selector	object	Label selector to match machines whose health will be exercised. Note: An empty selector will match all machines.

Property	Type	Description
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.
unhealthyConditions[]	object	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.

8.1.2. .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 Machine API Operator.

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 <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: 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

8.1.3. .spec.selector

Description

Label selector to match machines whose health will be exercised. Note: An empty selector will match all machines.

Type

object

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

8.1.4. .spec.selector.matchExpressions

Description

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

Type

array

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

8.1.6. .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

8.1.7. .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

Property	Type	Description
status	string	

Property	Type	Description
timeout	string	Expects an unsigned duration string of decimal numbers each with optional fraction and a unit suffix, eg "300ms", "1.5h" or "2h45m". Valid time units are "ns", "us" (or "µs"), "ms", "s", "m", "h".
type	string	

8.1.8. .status

Description

Most recently observed status of MachineHealthCheck resource

Type

object

Property	Type	Description
conditions	array	Conditions defines the current state of the MachineHealthCheck
conditions[]	object	Condition defines an observation of a Machine API resource operational state.
currentHealthy	integer	total number of machines counted by this machine health check
expectedMachines	integer	total number of machines counted by this machine health check
remediationsAllowed	integer	RemediationsAllowed is the number of further remediations allowed by this machine health check before maxUnhealthy short circuiting will be applied

8.1.9. .status.conditions

Description

Conditions defines the current state of the MachineHealthCheck

Type

array

8.1.10. .status.conditions[]

Description

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

Type

object

Required

- **type**

Property	Type	Description
lastTransitionTime	string	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	A human readable message indicating details about the transition. This field may be empty.
reason	string	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 not 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.

8.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machine.openshift.io/v1beta1/machinehealthchecks**
 - **GET**: list objects of kind MachineHealthCheck
- **/apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinehealthchecks**
 - **DELETE**: delete collection of MachineHealthCheck
 - **GET**: list objects of kind MachineHealthCheck
 - **POST**: create a MachineHealthCheck
- **/apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinehealthchecks/{name}**
 - **DELETE**: delete a MachineHealthCheck
 - **GET**: read the specified MachineHealthCheck
 - **PATCH**: partially update the specified MachineHealthCheck
 - **PUT**: replace the specified MachineHealthCheck
- **/apis/machine.openshift.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

8.2.1. /apis/machine.openshift.io/v1beta1/machinehealthchecks

HTTP method

GET

Description

list objects of kind MachineHealthCheck

Table 8.1. HTTP responses

HTTP code	Reponse body
200 - OK	MachineHealthCheckList schema
401 - Unauthorized	Empty

8.2.2. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinehealthc

HTTP method

DELETE

Description

delete collection of MachineHealthCheck

Table 8.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind MachineHealthCheck

Table 8.3. HTTP responses

HTTP code	Reponse body
200 - OK	MachineHealthCheckList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a MachineHealthCheck

Table 8.4. Query parameters

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

Table 8.5. Body parameters

Parameter	Type	Description
body	MachineHealthCheck schema	

Table 8.6. HTTP responses

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

8.2.3. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinehealthc

Table 8.7. Global path parameters

Parameter	Type	Description
name	string	name of the MachineHealthCheck

HTTP method

DELETE

Description

delete a MachineHealthCheck

Table 8.8. Query parameters

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

Table 8.9. HTTP responses

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

HTTP method

GET

Description

read the specified MachineHealthCheck

Table 8.10. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified MachineHealthCheck

Table 8.11. Query parameters

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

Table 8.12. HTTP responses

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

HTTP method**PUT****Description**

replace the specified MachineHealthCheck

Table 8.13. Query parameters

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

Table 8.14. Body parameters

Parameter	Type	Description
body	MachineHealthCheck schema	

Table 8.15. HTTP responses

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

8.2.4. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinehealthc

Table 8.16. Global path parameters

Parameter	Type	Description
name	string	name of the MachineHealthCheck

HTTP method**GET****Description**

read status of the specified MachineHealthCheck

Table 8.17. HTTP responses

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

HTTP method**PATCH****Description**

partially update status of the specified MachineHealthCheck

Table 8.18. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 8.19. HTTP responses

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

HTTP method**PUT****Description**

replace status of the specified MachineHealthCheck

Table 8.20. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 8.21. Body parameters

Parameter	Type	Description
body	MachineHealthCheck schema	

Table 8.22. HTTP responses

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

CHAPTER 9. MACHINE [MACHINE.OPENSIFT.IO/V1BETA1]

Description

Machine is the Schema for the machines API Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

9.1. SPECIFICATION

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

9.1.1. .spec

Description

MachineSpec defines the desired state of Machine

Type

object

Property	Type	Description
lifecycleHooks	object	LifecycleHooks allow users to pause operations on the machine at certain predefined points within the machine lifecycle.
metadata	object	ObjectMeta will autopopulate the Node created. Use this to indicate what labels, annotations, name prefix, etc., should be used when creating the Node.
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.
providerSpec	object	ProviderSpec details Provider-specific configuration to use during node creation.

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

9.1.2. .spec.lifecycleHooks

Description

LifecycleHooks allow users to pause operations on the machine at certain predefined points within the machine lifecycle.

Type

object

Property	Type	Description
preDrain	array	PreDrain hooks prevent the machine from being drained. This also blocks further lifecycle events, such as termination.
preDrain[]	object	LifecycleHook represents a single instance of a lifecycle hook
preTerminate	array	PreTerminate hooks prevent the machine from being terminated. PreTerminate hooks be actioned after the Machine has been drained.
preTerminate[]	object	LifecycleHook represents a single instance of a lifecycle hook

9.1.3. .spec.lifecycleHooks.preDrain

Description

PreDrain hooks prevent the machine from being drained. This also blocks further lifecycle events, such as termination.

Type

array

9.1.4. .spec.lifecycleHooks.preDrain[]

Description

LifecycleHook represents a single instance of a lifecycle hook

Type

object

Required

- **name**
- **owner**

Property	Type	Description
name	string	Name defines a unique name for the lifecycle hook. The name should be unique and descriptive, ideally 1-3 words, in CamelCase or it may be namespaced, eg. foo.example.com/CamelCase. Names must be unique and should only be managed by a single entity.
owner	string	Owner defines the owner of the lifecycle hook. This should be descriptive enough so that users can identify who/what is responsible for blocking the lifecycle. This could be the name of a controller (e.g. clusteroperator/etcd) or an administrator managing the hook.

9.1.5. .spec.lifecycleHooks.preTerminate

Description

PreTerminate hooks prevent the machine from being terminated. PreTerminate hooks be actioned after the Machine has been drained.

Type

array

9.1.6. .spec.lifecycleHooks.preTerminate[]

Description

LifecycleHook represents a single instance of a lifecycle hook

Type

object

Required

- **name**
- **owner**

Property	Type	Description
name	string	Name defines a unique name for the lifecycle hook. The name should be unique and descriptive, ideally 1-3 words, in CamelCase or it may be namespaced, eg. foo.example.com/CamelCase. Names must be unique and should only be managed by a single entity.
owner	string	Owner defines the owner of the lifecycle hook. This should be descriptive enough so that users can identify who/what is responsible for blocking the lifecycle. This could be the name of a controller (e.g. clusteroperator/etcd) or an administrator managing the hook.

9.1.7. .spec.metadata

Description

ObjectMeta will autopopulate the Node created. Use this to indicate what labels, annotations, name prefix, etc., should be used when creating the Node.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
annotations	object (string)	Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. More info: http://kubernetes.io/docs/user-guide/annotations
generateName	string	GenerateName is an optional prefix, used by the server, to generate a unique name ONLY IF the Name field has not been provided. If this field is used, the name returned to the client will be different than the name passed. This value will also be combined with a unique suffix. The provided value has the same validation rules as the Name field, and may be truncated by the length of the suffix required to make the value unique on the server. If this field is specified and the generated name exists, the server will NOT return a 409 - instead, it will either return 201 Created or 500 with Reason ServerTimeout indicating a unique name could not be found in the time allotted, and the client should retry (optionally after the time indicated in the Retry-After header). Applied only if Name is not specified. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#idempotency
labels	object (string)	Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services. More info: http://kubernetes.io/docs/user-guide/labels

Property	Type	Description
name	string	Name must be unique within a namespace. Is required when creating resources, although some resources may allow a client to request the generation of an appropriate name automatically. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names
namespace	string	Namespace defines the space within each name must be unique. An empty namespace is equivalent to the "default" namespace, but "default" is the canonical representation. Not all objects are required to be scoped to a namespace - the value of this field for those objects will be empty. Must be a DNS_LABEL. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/namespaces
ownerReferences	array	List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.
ownerReferences[]	object	OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

9.1.8. .spec.metadata.ownerReferences

Description

List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to

this controller, with the controller field set to true. There cannot be more than one managing controller.

Type

array

9.1.9. .spec.metadata.ownerReferences[]

Description

OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

Type

object

Required

- **apiVersion**
- **kind**
- **name**
- **uid**

Property	Type	Description
apiVersion	string	API version of the referent.
blockOwnerDeletion	boolean	If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is removed. See https://kubernetes.io/docs/concepts/architecture/garbage-collection/#foreground-deletion for how the garbage collector interacts with this field and enforces the foreground deletion. Defaults to false. To set this field, a user needs "delete" permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.
controller	boolean	If true, this reference points to the managing controller.

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

9.1.10. .spec.providerSpec

Description

ProviderSpec details Provider-specific configuration to use during node creation.

Type

object

Property	Type	Description
value	string	Value is an inlined, serialized representation of the resource configuration. It is recommended that providers maintain their own versioned API types that should be serialized/deserialized from this field, akin to component config.

9.1.11. .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

9.1.12. .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.

9.1.13. .status**Description**

MachineStatus defines the observed state of Machine

Type

object

Property	Type	Description
addresses	array	Addresses is a list of addresses assigned to the machine. Queried from cloud provider, if available.
addresses[]	object	NodeAddress contains information for the node's address.
conditions	array	Conditions defines the current state of the Machine

Property	Type	Description
conditions[]	object	Condition defines an observation of a Machine API resource operational state.
errorMessage	string	ErrorMessage 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. This field should not be set for transitive errors that a 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 of 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 to the Machine object and/or logged in the controller's output.

Property	Type	Description
errorReason	string	ErrorReason 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 a 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 of 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 to the Machine object and/or logged in the controller's output.
lastOperation	object	LastOperation describes the last-operation performed by the machine-controller. This API should be useful as a history in terms of the latest operation performed on the specific machine. It should also convey the state of the latest-operation for example if it is still on-going, failed or completed successfully.
lastUpdated	string	LastUpdated identifies when this status was last observed.
nodeRef	object	NodeRef will point to the corresponding Node if it exists.
phase	string	Phase represents the current phase of machine actuation. One of: Failed, Provisioning, Provisioned, Running, Deleting

Property	Type	Description
providerStatus	^^	ProviderStatus details a Provider-specific status. It is recommended that providers maintain their own versioned API types that should be serialized/deserialized from this field.

9.1.14. .status.addresses

Description

Addresses is a list of addresses assigned to the machine. Queried from cloud provider, if available.

Type

array

9.1.15. .status.addresses[]

Description

NodeAddress contains information for the node's address.

Type

object

Required

- **address**
- **type**

Property	Type	Description
address	string	The node address.
type	string	Node address type, one of Hostname, ExternalIP or InternalIP.

9.1.16. .status.conditions

Description

Conditions defines the current state of the Machine

Type

array

9.1.17. .status.conditions[]

Description

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

Type

object

Required

- **type**

Property	Type	Description
lastTransitionTime	string	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	A human readable message indicating details about the transition. This field may be empty.
reason	string	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 not 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.

9.1.18. .status.lastOperation

Description

LastOperation describes the last-operation performed by the machine-controller. This API should be useful as a history in terms of the latest operation performed on the specific machine. It should also convey the state of the latest-operation for example if it is still on-going, failed or completed successfully.

Type

object

Property	Type	Description
description	string	Description is the human-readable description of the last operation.
lastUpdated	string	LastUpdated is the timestamp at which LastOperation API was last-updated.
state	string	State is the current status of the last performed operation. E.g. Processing, Failed, Successful etc
type	string	Type is the type of operation which was last performed. E.g. Create, Delete, Update etc

9.1.19. .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.

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 <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: 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

9.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machine.openshift.io/v1beta1/machines**
 - **GET**: list objects of kind Machine
- **/apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machines**
 - **DELETE**: delete collection of Machine
 - **GET**: list objects of kind Machine
 - **POST**: create a Machine
- **/apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machines/{name}**
 - **DELETE**: delete a Machine
 - **GET**: read the specified Machine
 - **PATCH**: partially update the specified Machine
 - **PUT**: replace the specified Machine
- **/apis/machine.openshift.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

9.2.1. /apis/machine.openshift.io/v1beta1/machines

HTTP method

GET

Description

list objects of kind Machine

Table 9.1. HTTP responses

HTTP code	Reponse body
200 - OK	MachineList schema

HTTP code	Reponse body
401 - Unauthorized	Empty

9.2.2. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machines

HTTP method

DELETE

Description

delete collection of Machine

Table 9.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind Machine

Table 9.3. HTTP responses

HTTP code	Reponse body
200 - OK	MachineList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Machine

Table 9.4. Query parameters

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

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

Table 9.5. Body parameters

Parameter	Type	Description
body	Machine schema	

Table 9.6. HTTP responses

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

9.2.3. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machines/{name}

Table 9.7. Global path parameters

Parameter	Type	Description
name	string	name of the Machine

HTTP method**DELETE****Description**

delete a Machine

Table 9.8. Query parameters

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

Table 9.9. 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

Table 9.10. HTTP responses

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

HTTP method**PATCH****Description**

partially update the specified Machine

Table 9.11. Query parameters

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

Table 9.12. HTTP responses

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

HTTP method**PUT****Description**

replace the specified Machine

Table 9.13. Query parameters

Parameter	Type	Description
-----------	------	-------------

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

Table 9.14. Body parameters

Parameter	Type	Description
body	Machine schema	

Table 9.15. HTTP responses

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

9.2.4. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machines/{nam

Table 9.16. Global path parameters

Parameter	Type	Description
name	string	name of the Machine

HTTP method

GET

Description

read status of the specified Machine

Table 9.17. HTTP responses

HTTP code	Reponse body
200 - OK	Machine schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Machine

Table 9.18. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 9.19. HTTP responses

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

HTTP method**PUT****Description**

replace status of the specified Machine

Table 9.20. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 9.21. Body parameters

Parameter	Type	Description
body	Machine schema	

Table 9.22. HTTP responses

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

CHAPTER 10. MACHINESET

[MACHINE.OPENSIFT.IO/V1BETA1]

Description

MachineSet ensures that a specified number of machines replicas are running at any given time. Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

10.1. SPECIFICATION

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

10.1.1. .spec

Description

MachineSetSpec defines the desired state of MachineSet

Type

object

Property	Type	Description
deletePolicy	string	DeletePolicy defines the policy used to identify nodes to delete when downscaling. Defaults to "Random". Valid values are "Random, "Newest", "Oldest"
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)
replicas	integer	Replicas is the number of desired replicas. This is a pointer to distinguish between explicit zero and unspecified. Defaults to 1.
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/working-with-objects/labels/#label-selectors
template	object	Template is the object that describes the machine that will be created if insufficient replicas are detected.

10.1.2. .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.
matchExpressions[]	object	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchLabels	object (string)	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

10.1.3. .spec.selector.matchExpressions

Description

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

Type
array

10.1.4. .spec.selector.matchExpressions[]

Description

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

Type
object

Required

- **key**
- **operator**

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

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

10.1.5. .spec.template

Description

Template is the object that describes the machine that will be created if insufficient replicas are detected.

Type

object

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

10.1.6. .spec.template.metadata

Description

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

Type

object

Property	Type	Description
annotations	object (string)	Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. More info: http://kubernetes.io/docs/user-guide/annotations
generateName	string	GenerateName is an optional prefix, used by the server, to generate a unique name ONLY IF the Name field has not been provided. If this field is used, the name returned to the client will be different than the name passed. This value will also be combined with a unique suffix. The provided value has the same validation rules as the Name field, and may be truncated by the length of the suffix required to make the value unique on the server. If this field is specified and the generated name exists, the server will NOT return a 409 - instead, it will either return 201 Created or 500 with Reason ServerTimeout indicating a unique name could not be found in the time allotted, and the client should retry (optionally after the time indicated in the Retry-After header). Applied only if Name is not specified. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#idempotency
labels	object (string)	Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services. More info: http://kubernetes.io/docs/user-guide/labels

Property	Type	Description
name	string	Name must be unique within a namespace. Is required when creating resources, although some resources may allow a client to request the generation of an appropriate name automatically. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names
namespace	string	Namespace defines the space within each name must be unique. An empty namespace is equivalent to the "default" namespace, but "default" is the canonical representation. Not all objects are required to be scoped to a namespace - the value of this field for those objects will be empty. Must be a DNS_LABEL. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/namespaces
ownerReferences	array	List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.
ownerReferences[]	object	OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

10.1.7. .spec.template.metadata.ownerReferences

Description

List of objects depended by this object. If ALL objects in the list have been deleted, this object will

be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.

Type

array

10.1.8. .spec.template.metadata.ownerReferences[]

Description

OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

Type

object

Required

- **apiVersion**
- **kind**
- **name**
- **uid**

Property	Type	Description
apiVersion	string	API version of the referent.
blockOwnerDeletion	boolean	If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is removed. See https://kubernetes.io/docs/concepts/architecture/garbage-collection/#foreground-deletion for how the garbage collector interacts with this field and enforces the foreground deletion. Defaults to false. To set this field, a user needs "delete" permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.
controller	boolean	If true, this reference points to the managing controller.

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

10.1.9. .spec.template.spec

Description

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

Property	Type	Description
lifecycleHooks	object	LifecycleHooks allow users to pause operations on the machine at certain predefined points within the machine lifecycle.
metadata	object	ObjectMeta will autopopulate the Node created. Use this to indicate what labels, annotations, name prefix, etc., should be used when creating the Node.

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.
providerSpec	object	ProviderSpec details Provider-specific configuration to use during node creation.
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
taints[]	object	The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

10.1.10. .spec.template.spec.lifecycleHooks

Description

LifecycleHooks allow users to pause operations on the machine at certain predefined points within the machine lifecycle.

Type

object

Property	Type	Description
preDrain	array	PreDrain hooks prevent the machine from being drained. This also blocks further lifecycle events, such as termination.
preDrain[]	object	LifecycleHook represents a single instance of a lifecycle hook
preTerminate	array	PreTerminate hooks prevent the machine from being terminated. PreTerminate hooks be actioned after the Machine has been drained.
preTerminate[]	object	LifecycleHook represents a single instance of a lifecycle hook

10.1.11. .spec.template.spec.lifecycleHooks.preDrain

Description

PreDrain hooks prevent the machine from being drained. This also blocks further lifecycle events, such as termination.

Type

array

10.1.12. .spec.template.spec.lifecycleHooks.preDrain[]

Description

LifecycleHook represents a single instance of a lifecycle hook

Type

object

Required

- **name**
- **owner**

Property	Type	Description
name	string	Name defines a unique name for the lifecycle hook. The name should be unique and descriptive, ideally 1-3 words, in CamelCase or it may be namespaced, eg. foo.example.com/CamelCase. Names must be unique and should only be managed by a single entity.
owner	string	Owner defines the owner of the lifecycle hook. This should be descriptive enough so that users can identify who/what is responsible for blocking the lifecycle. This could be the name of a controller (e.g. clusteroperator/etcd) or an administrator managing the hook.

10.1.13. .spec.template.spec.lifecycleHooks.preTerminate

Description

PreTerminate hooks prevent the machine from being terminated. PreTerminate hooks be actioned after the Machine has been drained.

Type

array

10.1.14. .spec.template.spec.lifecycleHooks.preTerminate[]

Description

LifecycleHook represents a single instance of a lifecycle hook

Type

object

Required

- **name**
- **owner**

Property	Type	Description
----------	------	-------------

Property	Type	Description
name	string	Name defines a unique name for the lifecycle hook. The name should be unique and descriptive, ideally 1-3 words, in CamelCase or it may be namespaced, eg. foo.example.com/CamelCase. Names must be unique and should only be managed by a single entity.
owner	string	Owner defines the owner of the lifecycle hook. This should be descriptive enough so that users can identify who/what is responsible for blocking the lifecycle. This could be the name of a controller (e.g. clusteroperator/etcd) or an administrator managing the hook.

10.1.15. .spec.template.spec.metadata

Description

ObjectMeta will autopopulate the Node created. Use this to indicate what labels, annotations, name prefix, etc., should be used when creating the Node.

Type

object

Property	Type	Description
annotations	object (string)	Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. More info: http://kubernetes.io/docs/user-guide/annotations

Property	Type	Description
generateName	string	GenerateName is an optional prefix, used by the server, to generate a unique name ONLY IF the Name field has not been provided. If this field is used, the name returned to the client will be different than the name passed. This value will also be combined with a unique suffix. The provided value has the same validation rules as the Name field, and may be truncated by the length of the suffix required to make the value unique on the server. If this field is specified and the generated name exists, the server will NOT return a 409 - instead, it will either return 201 Created or 500 with Reason ServerTimeout indicating a unique name could not be found in the time allotted, and the client should retry (optionally after the time indicated in the Retry-After header). Applied only if Name is not specified. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#idempotency
labels	object (string)	Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services. More info: http://kubernetes.io/docs/user-guide/labels
name	string	Name must be unique within a namespace. Is required when creating resources, although some resources may allow a client to request the generation of an appropriate name automatically. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names

Property	Type	Description
namespace	string	Namespace defines the space within each name must be unique. An empty namespace is equivalent to the "default" namespace, but "default" is the canonical representation. Not all objects are required to be scoped to a namespace - the value of this field for those objects will be empty. Must be a DNS_LABEL. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/namespaces
ownerReferences	array	List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.
ownerReferences[]	object	OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

10.1.16. .spec.template.spec.metadata.ownerReferences

Description

List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.

Type

array

10.1.17. .spec.template.spec.metadata.ownerReferences[]

Description

OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

Type

object

Required

- **apiVersion**
- **kind**
- **name**
- **uid**

Property	Type	Description
apiVersion	string	API version of the referent.
blockOwnerDeletion	boolean	If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is removed. See https://kubernetes.io/docs/concepts/architecture/garbage-collection/#foreground-deletion for how the garbage collector interacts with this field and enforces the foreground deletion. Defaults to false. To set this field, a user needs "delete" permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.
controller	boolean	If true, this reference points to the managing controller.
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
uid	string	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids

10.1.18. .spec.template.spec.providerSpec

Description

ProviderSpec details Provider-specific configuration to use during node creation.

Type

object

Property	Type	Description
value	<code>..</code>	Value is an inlined, serialized representation of the resource configuration. It is recommended that providers maintain their own versioned API types that should be serialized/deserialized from this field, akin to component config.

10.1.19. .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

10.1.20. .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.

10.1.21. .status

Description

MachineSetStatus defines the observed state of MachineSet

Type

object

Property	Type	Description
availableReplicas	integer	The number of available replicas (ready for at least minReadySeconds) for this MachineSet.
conditions	array	Conditions defines the current state of the MachineSet
conditions[]	object	Condition defines an observation of a Machine API resource operational state.
errorMessage	string	

Property	Type	Description
errorReason	string	In the event that there is a terminal problem reconciling the replicas, both <code>ErrorReason</code> and <code>ErrorMessage</code> will be set. <code>ErrorReason</code> will be populated with a succinct value suitable for machine interpretation, while <code>ErrorMessage</code> 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 outages), but instead indicate that something is fundamentally wrong with the <code>MachineTemplate</code> '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 <code>Machines</code> can be added as events to the <code>MachineSet</code> object and/or logged in the controller's output.
fullyLabeledReplicas	integer	The number of replicas that have labels matching the labels of the machine template of the <code>MachineSet</code> .
observedGeneration	integer	<code>ObservedGeneration</code> reflects the generation of the most recently observed <code>MachineSet</code> .
readyReplicas	integer	The number of ready replicas for this <code>MachineSet</code> . A machine is considered ready when the node has been created and is "Ready".
replicas	integer	<code>Replicas</code> is the most recently observed number of replicas.

10.1.22. .status.conditions

Description

Conditions defines the current state of the MachineSet

Type

array

10.1.23. .status.conditions[]

Description

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

Type

object

Required

- **type**

Property	Type	Description
lastTransitionTime	string	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	A human readable message indicating details about the transition. This field may be empty.
reason	string	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 not 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.

10.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/machine.openshift.io/v1beta1/machinesets**
 - **GET**: list objects of kind MachineSet
- **/apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets**
 - **DELETE**: delete collection of MachineSet
 - **GET**: list objects of kind MachineSet
 - **POST**: create a MachineSet
- **/apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets/{name}**
 - **DELETE**: delete a MachineSet
 - **GET**: read the specified MachineSet
 - **PATCH**: partially update the specified MachineSet
 - **PUT**: replace the specified MachineSet
- **/apis/machine.openshift.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/machine.openshift.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

10.2.1. /apis/machine.openshift.io/v1beta1/machinesets

HTTP method**GET****Description**

list objects of kind MachineSet

Table 10.1. HTTP responses

HTTP code	Reponse body
200 - OK	MachineSetList schema
401 - Unauthorized	Empty

10.2.2. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets**HTTP method****DELETE****Description**

delete collection of MachineSet

Table 10.2. HTTP responses

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

HTTP method**GET****Description**

list objects of kind MachineSet

Table 10.3. HTTP responses

HTTP code	Reponse body
200 - OK	MachineSetList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a MachineSet

Table 10.4. Query parameters

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

Table 10.5. Body parameters

Parameter	Type	Description
body	MachineSet schema	

Table 10.6. HTTP responses

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

10.2.3. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets/{i

Table 10.7. Global path parameters

Parameter	Type	Description
name	string	name of the MachineSet

HTTP method

DELETE

Description

delete a MachineSet

Table 10.8. Query parameters

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

Table 10.9. HTTP responses

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

HTTP method

GET

Description

read the specified MachineSet

Table 10.10. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified MachineSet

Table 10.11. Query parameters

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

Table 10.12. HTTP responses

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

HTTP method**PUT****Description**

replace the specified MachineSet

Table 10.13. Query parameters

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

Table 10.14. Body parameters

Parameter	Type	Description
body	MachineSet schema	

Table 10.15. HTTP responses

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

10.2.4. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets/{}

Table 10.16. Global path parameters

Parameter	Type	Description
name	string	name of the MachineSet

HTTP method

GET

Description

read scale of the specified MachineSet

Table 10.17. HTTP responses

HTTP code	Reponse body
200 - OK	Scale schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update scale of the specified MachineSet

Table 10.18. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 10.19. HTTP responses

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

HTTP method**PUT****Description**

replace scale of the specified MachineSet

Table 10.20. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 10.21. Body parameters

Parameter	Type	Description
body	Scale schema	

Table 10.22. HTTP responses

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

10.2.5. /apis/machine.openshift.io/v1beta1/namespaces/{namespace}/machinesets/{i

Table 10.23. Global path parameters

Parameter	Type	Description
name	string	name of the MachineSet

HTTP method

GET

Description

read status of the specified MachineSet

Table 10.24. HTTP responses

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

HTTP method**PATCH****Description**

partially update status of the specified MachineSet

Table 10.25. Query parameters

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

Table 10.26. HTTP responses

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

HTTP method**PUT****Description**

replace status of the specified MachineSet

Table 10.27. Query parameters

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

Table 10.28. Body parameters

Parameter	Type	Description
body	MachineSet schema	

Table 10.29. HTTP responses

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