



OpenShift Container Platform 4.18

Provisioning APIs

Reference guide for provisioning APIs

OpenShift Container Platform 4.18 Provisioning APIs

Reference guide for provisioning APIs

Legal Notice

Copyright © 2025 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 provisioning API objects and their detailed specifications.

Table of Contents

CHAPTER 1. PROVISIONING APIS	6
1.1. BMCEVENTSUBSCRIPTION [METAL3.IO/V1ALPHA1]	6
1.2. BAREMETALHOST [METAL3.IO/V1ALPHA1]	6
1.3. DATAIMAGE [METAL3.IO/V1ALPHA1]	6
1.4. FIRMWARESCHEMA [METAL3.IO/V1ALPHA1]	6
1.5. HARDWAREDATA [METAL3.IO/V1ALPHA1]	6
1.6. HOSTFIRMWARECOMPONENTS [METAL3.IO/V1ALPHA1]	6
1.7. HOSTFIRMWARESETTINGS [METAL3.IO/V1ALPHA1]	6
1.8. METAL3REMIEDIATION [INFRASTRUCTURE.CLUSTER.X-K8S.IO/VIBETA1]	7
1.9. METAL3REMIEDIATIONTEMPLATE [INFRASTRUCTURE.CLUSTER.X-K8S.IO/VIBETA1]	7
1.10. PREPROVISIONINGIMAGE [METAL3.IO/V1ALPHA1]	7
1.11. PROVISIONING [METAL3.IO/V1ALPHA1]	7
CHAPTER 2. BMCEVENTSUBSCRIPTION [METAL3.IO/V1ALPHA1]	8
2.1. SPECIFICATION	8
2.1.1. .spec	8
2.1.2. .spec.httpHeadersRef	9
2.1.3. .status	9
2.2. API ENDPOINTS	10
2.2.1. /apis/metal3.io/v1alpha1/bmceventsubscriptions	10
2.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions	10
2.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions/{name}	12
2.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions/{name}/status	15
CHAPTER 3. BAREMETALHOST [METAL3.IO/V1ALPHA1]	19
3.1. SPECIFICATION	19
3.1.1. .spec	19
3.1.2. .spec.bmc	23
3.1.3. .spec.consumerRef	24
3.1.4. .spec.customDeploy	25
3.1.5. .spec.firmware	26
3.1.6. .spec.image	26
3.1.7. .spec.metaData	27
3.1.8. .spec.networkData	27
3.1.9. .spec.raid	28
3.1.10. .spec.rootDeviceHints	28
3.1.11. .spec.taints	30
3.1.12. .spec.taints[]	30
3.1.13. .spec.userData	30
3.1.14. .status	31
3.1.15. .status.goodCredentials	32
3.1.16. .status.goodCredentials.credentials	33
3.1.17. .status.hardware	33
3.1.18. .status.hardware.cpu	34
3.1.19. .status.hardware.firmware	34
3.1.20. .status.hardware.firmware.bios	35
3.1.21. .status.hardware.nics	35
3.1.22. .status.hardware.nics[]	35
3.1.23. .status.hardware.nics[].vlans	36
3.1.24. .status.hardware.nics[].vlans[]	36
3.1.25. .status.hardware.storage	36

3.1.26. .status.hardware.storage[]	37
3.1.27. .status.hardware.systemVendor	38
3.1.28. .status.operationHistory	38
3.1.29. .status.operationHistory.deprovision	39
3.1.30. .status.operationHistory.inspect	39
3.1.31. .status.operationHistory.provision	39
3.1.32. .status.operationHistory.register	40
3.1.33. .status.provisioning	40
3.1.34. .status.provisioning.customDeploy	41
3.1.35. .status.provisioning.firmware	41
3.1.36. .status.provisioning.image	42
3.1.37. .status.provisioning.raid	43
3.1.38. .status.provisioning.rootDeviceHints	43
3.1.39. .status.triedCredentials	44
3.1.40. .status.triedCredentials.credentials	45
3.2. API ENDPOINTS	45
3.2.1. /apis/metal3.io/v1alpha1/baremetalhosts	46
3.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts	46
3.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts/{name}	48
3.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts/{name}/status	51
CHAPTER 4. DATAIMAGE [METAL3.IO/V1ALPHA1]	54
4.1. SPECIFICATION	54
4.1.1. .spec	54
4.1.2. .status	55
4.1.3. .status.attachedImage	55
4.1.4. .status.error	55
4.2. API ENDPOINTS	56
4.2.1. /apis/metal3.io/v1alpha1/dataimages	56
4.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages	57
4.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages/{name}	58
4.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages/{name}/status	61
CHAPTER 5. FIRMWARESCHEMA [METAL3.IO/V1ALPHA1]	65
5.1. SPECIFICATION	65
5.1.1. .spec	65
5.1.2. .spec.schema	66
5.1.3. .spec.schema{}	66
5.2. API ENDPOINTS	67
5.2.1. /apis/metal3.io/v1alpha1/firmwareschemas	67
5.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/firmwareschemas	68
5.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/firmwareschemas/{name}	69
CHAPTER 6. HARDWAREDATA [METAL3.IO/V1ALPHA1]	73
6.1. SPECIFICATION	73
6.1.1. .spec	73
6.1.2. .spec.hardware	74
6.1.3. .spec.hardware.cpu	74
6.1.4. .spec.hardware.firmware	75
6.1.5. .spec.hardware.firmware.bios	75
6.1.6. .spec.hardware.nics	75
6.1.7. .spec.hardware.nics[]	76
6.1.8. .spec.hardware.nics[].vlans	76
6.1.9. .spec.hardware.nics[].vlans[]	76

6.1.10. .spec.hardware.storage	77
6.1.11. .spec.hardware.storage[]	77
6.1.12. .spec.hardware.systemVendor	78
6.2. API ENDPOINTS	78
6.2.1. /apis/metal3.io/v1alpha1/hardwaredata	79
6.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hardwaredata	79
6.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hardwaredata/{name}	81
CHAPTER 7. HOSTFIRMWARECOMPONENTS [METAL3.IO/V1ALPHA1]	85
7.1. SPECIFICATION	85
7.1.1. .spec	86
7.1.2. .spec.updates	86
7.1.3. .spec.updates[]	86
7.1.4. .status	86
7.1.5. .status.components	87
7.1.6. .status.components[]	87
7.1.7. .status.conditions	88
7.1.8. .status.conditions[]	88
7.1.9. .status.updates	89
7.1.10. .status.updates[]	89
7.2. API ENDPOINTS	90
7.2.1. /apis/metal3.io/v1alpha1/hostfirmwarecomponents	90
7.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents	91
7.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents/{name}	92
7.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents/{name}/status	95
CHAPTER 8. HOSTFIRMWARESETTINGS [METAL3.IO/V1ALPHA1]	99
8.1. SPECIFICATION	99
8.1.1. .spec	100
8.1.2. .status	100
8.1.3. .status.conditions	101
8.1.4. .status.conditions[]	101
8.1.5. .status.schema	102
8.2. API ENDPOINTS	103
8.2.1. /apis/metal3.io/v1alpha1/hostfirmwaresettings	103
8.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings	103
8.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings/{name}	105
8.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings/{name}/status	108
CHAPTER 9. METAL3REMEDICATION [INFRASTRUCTURE.CLUSTER.X-K8S.IO/V1BETA1]	112
9.1. SPECIFICATION	112
9.1.1. .spec	113
9.1.2. .spec.strategy	113
9.1.3. .status	113
9.2. API ENDPOINTS	114
9.2.1. /apis/infrastructure.cluster.x-k8s.io/v1beta1/metal3remediations	114
9.2.2. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations	115
9.2.3. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations/{name}	116
9.2.4. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations/{name}/status	119
CHAPTER 10. METAL3REMEDICATIONTEMPLATE [INFRASTRUCTURE.CLUSTER.X-K8S.IO/V1BETA1]	123
10.1. SPECIFICATION	123

10.1.1. .spec	124
10.1.2. .spec.template	124
10.1.3. .spec.template.spec	124
10.1.4. .spec.template.spec.strategy	125
10.1.5. .status	125
10.1.6. .status.status	125
10.2. API ENDPOINTS	126
10.2.1. /apis/infrastructure.cluster.x-k8s.io/v1beta1/metal3remediationtemplates	126
10.2.2. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates	127
10.2.3. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates/{name}	128
10.2.4. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates/{name}/status	131
CHAPTER 11. PREPROVISIONINGIMAGE [METAL3.IO/V1ALPHA1]	135
11.1. SPECIFICATION	135
11.1.1. .spec	136
11.1.2. .status	136
11.1.3. .status.conditions	137
11.1.4. .status.conditions[]	137
11.1.5. .status.networkData	138
11.2. API ENDPOINTS	139
11.2.1. /apis/metal3.io/v1alpha1/preprovisioningimages	139
11.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages	140
11.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages/{name}	141
11.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages/{name}/status	144
CHAPTER 12. PROVISIONING [METAL3.IO/V1ALPHA1]	148
12.1. SPECIFICATION	148
12.1.1. .spec	149
12.1.2. .spec.preProvisioningOSDownloadURLs	153
12.1.3. .status	154
12.1.4. .status.conditions	154
12.1.5. .status.conditions[]	155
12.1.6. .status.generations	155
12.1.7. .status.generations[]	155
12.2. API ENDPOINTS	156
12.2.1. /apis/metal3.io/v1alpha1/provisionings	156
12.2.2. /apis/metal3.io/v1alpha1/provisionings/{name}	158
12.2.3. /apis/metal3.io/v1alpha1/provisionings/{name}/status	161

CHAPTER 1. PROVISIONING APIS

1.1. BMCEVENTSUBSCRIPTION [METAL3.IO/V1ALPHA1]

Description

BMCEventSubscription is the Schema for the fast eventing API

Type

object

1.2. BAREMETALHOST [METAL3.IO/V1ALPHA1]

Description

BareMetalHost is the Schema for the baremetalhosts API

Type

object

1.3. DATAIMAGE [METAL3.IO/V1ALPHA1]

Description

DataImage is the Schema for the dataimages API.

Type

object

1.4. FIRMWARESCHEMA [METAL3.IO/V1ALPHA1]

Description

FirmwareSchema is the Schema for the firmwareschemas API.

Type

object

1.5. HARDWAREDATA [METAL3.IO/V1ALPHA1]

Description

HardwareData is the Schema for the hardwaredata API.

Type

object

1.6. HOSTFIRMWARECOMPONENTS [METAL3.IO/V1ALPHA1]

Description

HostFirmwareComponents is the Schema for the hostfirmwarecomponents API.

Type

object

1.7. HOSTFIRMWARESETTINGS [METAL3.IO/V1ALPHA1]

Description

HostFirmwareSettings is the Schema for the hostfirmwaresettings API.

Type

object

1.8. METAL3REMEDICATION [INFRASTRUCTURE.CLUSTER.X-K8S.IO/V1BETA1]

Description

Metal3Remediation is the Schema for the metal3remediations API.

Type

object

1.9. METAL3REMEDICATIONTEMPLATE [INFRASTRUCTURE.CLUSTER.X-K8S.IO/V1BETA1]

Description

Metal3RemediationTemplate is the Schema for the metal3remediationtemplates API.

Type

object

1.10. PREPROVISIONINGIMAGE [METAL3.IO/V1ALPHA1]

Description

PreprovisioningImage is the Schema for the preprovisioningimages API.

Type

object

1.11. PROVISIONING [METAL3.IO/V1ALPHA1]

Description

Provisioning contains configuration used by the Provisioning service (Ironic) to provision baremetal hosts. Provisioning is created by the OpenShift installer using admin or user provided information about the provisioning network and the NIC on the server that can be used to PXE boot it. This CR is a singleton, created by the installer and currently only consumed by the cluster-baremetal-operator to bring up and update containers in a metal3 cluster.

Type

object

CHAPTER 2. BMCEVENTSUBSCRIPTION [METAL3.IO/V1ALPHA1]

Description

BMCEventSubscription is the Schema for the fast eventing API

Type

object

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	
status	object	

2.1.1. .spec

Description

Type
object

Property	Type	Description
context	string	Arbitrary user-provided context for the event
destination	string	A webhook URL to send events to
hostName	string	A reference to a BareMetalHost
httpHeadersRef	object	A secret containing HTTP headers which should be passed along to the Destination when making a request

2.1.2. .spec.httpHeadersRef

Description

A secret containing HTTP headers which should be passed along to the Destination when making a request

Type
object

Property	Type	Description
name	string	name is unique within a namespace to reference a secret resource.
namespace	string	namespace defines the space within which the secret name must be unique.

2.1.3. .status

Description

Type
object

Property	Type	Description
error	string	
subscriptionID	string	

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/metal3.io/v1alpha1/bmceventsubscriptions**
 - **GET**: list objects of kind BMCEventSubscription
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions**
 - **DELETE**: delete collection of BMCEventSubscription
 - **GET**: list objects of kind BMCEventSubscription
 - **POST**: create a BMCEventSubscription
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions/{name}**
 - **DELETE**: delete a BMCEventSubscription
 - **GET**: read the specified BMCEventSubscription
 - **PATCH**: partially update the specified BMCEventSubscription
 - **PUT**: replace the specified BMCEventSubscription
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions/{name}/status**
 - **GET**: read status of the specified BMCEventSubscription
 - **PATCH**: partially update status of the specified BMCEventSubscription
 - **PUT**: replace status of the specified BMCEventSubscription

2.2.1. /apis/metal3.io/v1alpha1/bmceventsubscriptions

HTTP method

GET

Description

list objects of kind BMCEventSubscription

Table 2.1. HTTP responses

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

2.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions

HTTP method

DELETE

Description

delete collection of BMCEventSubscription

Table 2.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind BMCEventSubscription

Table 2.3. HTTP responses

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

HTTP method

POST

Description

create a BMCEventSubscription

Table 2.4. Query parameters

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

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 2.5. Body parameters

Parameter	Type	Description
body	BMCEventSubscription schema	

Table 2.6. HTTP responses

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

2.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions/{n

Table 2.7. Global path parameters

Parameter	Type	Description
name	string	name of the BMCEventSubscription

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

delete a BMCEventSubscription

Table 2.8. Query parameters

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

Table 2.9. HTTP responses

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

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

read the specified BMCEventSubscription

Table 2.10. HTTP responses

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

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

partially update the specified BMCEventSubscription

Table 2.11. Query parameters

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

Table 2.12. HTTP responses

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

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

replace the specified BMCEventSubscription

Table 2.13. 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 2.14. Body parameters

Parameter	Type	Description
body	BMCEventSubscription schema	

Table 2.15. HTTP responses

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

2.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/bmceventsubscriptions/{n

Table 2.16. Global path parameters

Parameter	Type	Description
name	string	name of the BMCEventSubscription

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

read status of the specified BMCEventSubscription

Table 2.17. HTTP responses

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

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

partially update status of the specified BMCEventSubscription

Table 2.18. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <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 2.19. HTTP responses

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

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

replace status of the specified BMCEventSubscription

Table 2.20. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <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 2.21. Body parameters

Parameter	Type	Description
body	BMCEventSubscription schema	

Table 2.22. HTTP responses

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

CHAPTER 3. BAREMETALHOST [METAL3.IO/V1ALPHA1]

Description

BareMetalHost is the Schema for the baremetalhosts API

Type

object

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	BareMetalHostSpec defines the desired state of BareMetalHost.
status	object	BareMetalHostStatus defines the observed state of BareMetalHost.

3.1.1. .spec

Description

BareMetalHostSpec defines the desired state of BareMetalHost.

Type

object

Required

- **online**

Property	Type	Description
architecture	string	CPU architecture of the host, e.g. "x86_64" or "aarch64". If unset, eventually populated by inspection.
automatedCleaningMode	string	When set to disabled, automated cleaning will be skipped during provisioning and deprovisioning.
bmc	object	How do we connect to the BMC (Baseboard Management Controller) on the host?
bootMACAddress	string	The MAC address of the NIC used for provisioning the host. In case of network boot, this is the MAC address of the PXE booting interface. The MAC address of the BMC must never be used here!
bootMode	string	Select the method of initializing the hardware during boot. Defaults to UEFI. Legacy boot should only be used for hardware that does not support UEFI correctly. Set to UEFISecureBoot to turn secure boot on automatically after provisioning.
consumerRef	object	ConsumerRef can be used to store information about something that is using a host. When it is not empty, the host is considered "in use". The common use case is a link to a Machine resource when the host is used by Cluster API.

Property	Type	Description
customDeploy	object	A custom deploy procedure. This is an advanced feature that allows using a custom deploy step provided by a site-specific deployment ramdisk. Most users will want to use "image" instead. Setting this field triggers provisioning.
description	string	Description is a human-entered text used to help identify the host.
externallyProvisioned	boolean	ExternallyProvisioned means something else has provisioned the image running on the host, and the operator should only manage the power status. This field is used for integration with already provisioned hosts and when pivoting hosts between clusters. If unsure, leave this field as false.
firmware	object	Firmware (BIOS) configuration for bare metal server. If set, the requested settings will be applied before the host is provisioned. Only some vendor drivers support this field. An alternative is to use HostFirmwareSettings resources that allow changing arbitrary values and support the generic Redfish-based drivers.
hardwareProfile	string	What is the name of the hardware profile for this host? Hardware profiles are deprecated and should not be used. Use the separate fields Architecture and RootDeviceHints instead. Set to "empty" to prepare for the future version of the API without hardware profiles.
image	object	Image holds the details of the image to be provisioned. Populating the image will cause the host to start provisioning.

Property	Type	Description
metaData	object	MetaData holds the reference to the Secret containing host metadata which is passed to the Config Drive. By default, metadata will be generated for the host, so most users do not need to set this field.
networkData	object	NetworkData holds the reference to the Secret containing network configuration which is passed to the Config Drive and interpreted by the first boot software such as cloud-init.
online	boolean	Should the host be powered on? If the host is currently in a stable state (e.g. provisioned), its power state will be forced to match this value.
preprovisioningNetworkData Name	string	PreprovisioningNetworkDataName is the name of the Secret in the local namespace containing network configuration which is passed to the preprovisioning image, and to the Config Drive if not overridden by specifying NetworkData.
raid	object	RAID configuration for bare metal server. If set, the RAID settings will be applied before the host is provisioned. If not, the current settings will not be modified. Only one of the sub-fields hardwareRAIDVolumes and softwareRAIDVolumes can be set at the same time.
rootDeviceHints	object	Provide guidance about how to choose the device for the image being provisioned. The default is currently to use /dev/sda as the root device.

Property	Type	Description
taints	array	Taints is the full, authoritative list of taints to apply to the corresponding Machine. This list will overwrite any modifications made to the Machine on an ongoing basis.
taints[]	object	The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.
userData	object	UserData holds the reference to the Secret containing the user data which is passed to the Config Drive and interpreted by the first-boot software such as cloud-init. The format of user data is specific to the first-boot software.

3.1.2. .spec.bmc

Description

How do we connect to the BMC (Baseboard Management Controller) on the host?

Type

object

Required

- **address**
- **credentialsName**

Property	Type	Description
address	string	Address holds the URL for accessing the controller on the network. The scheme part designates the driver to use with the host.
credentialsName	string	The name of the secret containing the BMC credentials (requires keys "username" and "password").

Property	Type	Description
disableCertificateVerification	boolean	DisableCertificateVerification disables verification of server certificates when using HTTPS to connect to the BMC. This is required when the server certificate is self-signed, but is insecure because it allows a man-in-the-middle to intercept the connection.

3.1.3. .spec.consumerRef

Description

ConsumerRef can be used to store information about something that is using a host. When it is not empty, the host is considered "in use". The common use case is a link to a Machine resource when the host is used by Cluster API.

Type

object

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

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

3.1.4. .spec.customDeploy

Description

A custom deploy procedure. This is an advanced feature that allows using a custom deploy step provided by a site-specific deployment ramdisk. Most users will want to use "image" instead. Setting this field triggers provisioning.

Type

object

Required

- **method**

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

Property	Type	Description
method	string	Custom deploy method name. This name is specific to the deploy ramdisk used. If you don't have a custom deploy ramdisk, you shouldn't use CustomDeploy.

3.1.5. .spec.firmware

Description

Firmware (BIOS) configuration for bare metal server. If set, the requested settings will be applied before the host is provisioned. Only some vendor drivers support this field. An alternative is to use HostFirmwareSettings resources that allow changing arbitrary values and support the generic Redfish-based drivers.

Type

object

Property	Type	Description
simultaneousMultithreadingEnabled	boolean	Allows a single physical processor core to appear as several logical processors.
sriovEnabled	boolean	SR-IOV support enables a hypervisor to create virtual instances of a PCI-express device, potentially increasing performance.
virtualizationEnabled	boolean	Supports the virtualization of platform hardware.

3.1.6. .spec.image

Description

Image holds the details of the image to be provisioned. Populating the image will cause the host to start provisioning.

Type

object

Required

- **url**

Property	Type	Description
checksum	string	Checksum is the checksum for the image. Required for all formats except for "live-iso".
checksumType	string	ChecksumType is the checksum algorithm for the image, e.g md5, sha256 or sha512. The special value "auto" can be used to detect the algorithm from the checksum. If missing, MD5 is used. If in doubt, use "auto".
format	string	Format contains the format of the image (raw, qcow2, ...). When set to "live-iso", an ISO 9660 image referenced by the url will be live-booted and not deployed to disk.
url	string	URL is a location of an image to deploy.

3.1.7. .spec.metaData

Description

MetaData holds the reference to the Secret containing host metadata which is passed to the Config Drive. By default, metadata will be generated for the host, so most users do not need to set this field.

Type

object

Property	Type	Description
name	string	name is unique within a namespace to reference a secret resource.
namespace	string	namespace defines the space within which the secret name must be unique.

3.1.8. .spec.networkData

Description

NetworkData holds the reference to the Secret containing network configuration which is passed to the Config Drive and interpreted by the first boot software such as cloud-init.

Type

object

Property	Type	Description
name	string	name is unique within a namespace to reference a secret resource.
namespace	string	namespace defines the space within which the secret name must be unique.

3.1.9. .spec.raid

Description

RAID configuration for bare metal server. If set, the RAID settings will be applied before the host is provisioned. If not, the current settings will not be modified. Only one of the sub-fields hardwareRAIDVolumes and softwareRAIDVolumes can be set at the same time.

Type

object

Property	Type	Description
hardwareRAIDVolumes	``	The list of logical disks for hardware RAID, if rootDeviceHints isn't used, first volume is root volume. You can set the value of this field to <code>[]</code> to clear all the hardware RAID configurations.
softwareRAIDVolumes	``	The list of logical disks for software RAID, if rootDeviceHints isn't used, first volume is root volume. If HardwareRAIDVolumes is set this item will be invalid. The number of created Software RAID devices must be 1 or 2. If there is only one Software RAID device, it has to be a RAID-1. If there are two, the first one has to be a RAID-1, while the RAID level for the second one can be 0, 1, or 1+0. As the first RAID device will be the deployment device, enforcing a RAID-1 reduces the risk of ending up with a non-booting host in case of a disk failure. Software RAID will always be deleted.

3.1.10. .spec.rootDeviceHints

Description

Provide guidance about how to choose the device for the image being provisioned. The default is currently to use `/dev/sda` as the root device.

Type

object

Property	Type	Description
deviceName	string	A Linux device name like <code>/dev/vda</code> , or a by-path link to it like <code>/dev/disk/by-path/pci-0000:01:00.0-scsi-0:2:0:0</code> . The hint must match the actual value exactly.
hctl	string	A SCSI bus address like <code>0:0:0:0</code> . The hint must match the actual value exactly.
minSizeGigabytes	integer	The minimum size of the device in Gigabytes.
model	string	A vendor-specific device identifier. The hint can be a substring of the actual value.
rotational	boolean	True if the device should use spinning media, false otherwise.
serialNumber	string	Device serial number. The hint must match the actual value exactly.
vendor	string	The name of the vendor or manufacturer of the device. The hint can be a substring of the actual value.
wwn	string	Unique storage identifier. The hint must match the actual value exactly.
wwnVendorExtension	string	Unique vendor storage identifier. The hint must match the actual value exactly.
wwnWithExtension	string	Unique storage identifier with the vendor extension appended. The hint must match the actual value exactly.

3.1.11. .spec.taints

Description

Taints is the full, authoritative list of taints to apply to the corresponding Machine. This list will overwrite any modifications made to the Machine on an ongoing basis.

Type

array

3.1.12. .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.

3.1.13. .spec.userData

Description

UserData holds the reference to the Secret containing the user data which is passed to the Config Drive and interpreted by the first-boot software such as cloud-init. The format of user data is specific to the first-boot software.

Type

object

Property	Type	Description
name	string	name is unique within a namespace to reference a secret resource.
namespace	string	namespace defines the space within which the secret name must be unique.

3.1.14. .status

Description

BareMetalHostStatus defines the observed state of BareMetalHost.

Type

object

Required

- **errorCount**
- **errorMessage**
- **operationalStatus**
- **poweredOn**
- **provisioning**

Property	Type	Description
errorCount	integer	ErrorCount records how many times the host has encountered an error since the last successful operation
errorMessage	string	The last error message reported by the provisioning subsystem.
errorType	string	ErrorType indicates the type of failure encountered when the OperationalStatus is OperationalStatusError
goodCredentials	object	The last credentials we were able to validate as working.

Property	Type	Description
hardware	object	The hardware discovered to exist on the host. This field will be removed in the next API version in favour of the separate HardwareData resource.
hardwareProfile	string	The name of the profile matching the hardware details. Hardware profiles are deprecated and should not be relied on.
lastUpdated	string	LastUpdated identifies when this status was last observed.
operationHistory	object	OperationHistory holds information about operations performed on this host.
operationalStatus	string	OperationalStatus holds the status of the host
poweredOn	boolean	The currently detected power state of the host. This field may get briefly out of sync with the actual state of the hardware while provisioning processes are running.
provisioning	object	Information tracked by the provisioner.
triedCredentials	object	The last credentials we sent to the provisioning backend.

3.1.15. .status.goodCredentials

Description

The last credentials we were able to validate as working.

Type

object

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

Property	Type	Description
credentials	object	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
credentialsVersion	string	

3.1.16. .status.goodCredentials.credentials

Description

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

Type

object

Property	Type	Description
name	string	name is unique within a namespace to reference a secret resource.
namespace	string	namespace defines the space within which the secret name must be unique.

3.1.17. .status.hardware

Description

The hardware discovered to exist on the host. This field will be removed in the next API version in favour of the separate HardwareData resource.

Type

object

Property	Type	Description
cpu	object	Details of the CPU(s) in the system.
firmware	object	System firmware information.
hostname	string	
nics	array	List of network interfaces for the host.

Property	Type	Description
nics[]	object	NIC describes one network interface on the host.
ramMebibytes	integer	The host's amount of memory in Mebibytes.
storage	array	List of storage (disk, SSD, etc.) available to the host.
storage[]	object	Storage describes one storage device (disk, SSD, etc.) on the host.
systemVendor	object	System vendor information.

3.1.18. .status.hardware.cpu

Description

Details of the CPU(s) in the system.

Type

object

Property	Type	Description
arch	string	
clockMegahertz	number	ClockSpeed is a clock speed in MHz
count	integer	
flags	array (string)	
model	string	

3.1.19. .status.hardware.firmware

Description

System firmware information.

Type

object

Property	Type	Description
bios	object	The BIOS for this firmware

3.1.20. .status.hardware.firmware.bios

Description

The BIOS for this firmware

Type

object

Property	Type	Description
date	string	The release/build date for this BIOS
vendor	string	The vendor name for this BIOS
version	string	The version of the BIOS

3.1.21. .status.hardware.nics

Description

List of network interfaces for the host.

Type

array

3.1.22. .status.hardware.nics[]

Description

NIC describes one network interface on the host.

Type

object

Property	Type	Description
ip	string	The IP address of the interface. This will be an IPv4 or IPv6 address if one is present. If both IPv4 and IPv6 addresses are present in a dual-stack environment, two nics will be output, one with each IP.
mac	string	The device MAC address

Property	Type	Description
model	string	The vendor and product IDs of the NIC, e.g. "0x8086 0x1572"
name	string	The name of the network interface, e.g. "en0"
pxe	boolean	Whether the NIC is PXE Bootable
speedGbps	integer	The speed of the device in Gigabits per second
vlanId	integer	The untagged VLAN ID
vlangs	array	The VLANs available
vlangs[]	object	VLAN represents the name and ID of a VLAN.

3.1.23. `.status.hardware.nics[].vlangs`

Description

The VLANs available

Type

array

3.1.24. `.status.hardware.nics[].vlangs[]`

Description

VLAN represents the name and ID of a VLAN.

Type

object

Property	Type	Description
id	integer	VLANID is a 12-bit 802.1Q VLAN identifier
name	string	

3.1.25. `.status.hardware.storage`

Description

List of storage (disk, SSD, etc.) available to the host.

Type

array

3.1.26. .status.hardware.storage[]

Description

Storage describes one storage device (disk, SSD, etc.) on the host.

Type

object

Property	Type	Description
alternateNames	array (string)	A list of alternate Linux device names of the disk, e.g. <code>"/dev/sda"</code> . Note that this list is not exhaustive, and names may not be stable across reboots.
hctl	string	The SCSI location of the device
model	string	Hardware model
name	string	A Linux device name of the disk, e.g. <code>"/dev/disk/by-path/pci-0000:01:00.0-scsi-0:2:0:0"</code> . This will be a name that is stable across reboots if one is available.
rotational	boolean	Whether this disk represents rotational storage. This field is not recommended for usage, please prefer using 'Type' field instead, this field will be deprecated eventually.
serialNumber	string	The serial number of the device
sizeBytes	integer	The size of the disk in Bytes
type	string	Device type, one of: HDD, SSD, NVME.
vendor	string	The name of the vendor of the device
wwn	string	The WWN of the device
wwnVendorExtension	string	The WWN Vendor extension of the device

Property	Type	Description
wwnWithExtension	string	The WWN with the extension

3.1.27. .status.hardware.systemVendor

Description

System vendor information.

Type

object

Property	Type	Description
manufacturer	string	
productName	string	
serialNumber	string	

3.1.28. .status.operationHistory

Description

OperationHistory holds information about operations performed on this host.

Type

object

Property	Type	Description
deprovision	object	OperationMetric contains metadata about an operation (inspection, provisioning, etc.) used for tracking metrics.
inspect	object	OperationMetric contains metadata about an operation (inspection, provisioning, etc.) used for tracking metrics.
provision	object	OperationMetric contains metadata about an operation (inspection, provisioning, etc.) used for tracking metrics.

Property	Type	Description
register	object	OperationMetric contains metadata about an operation (inspection, provisioning, etc.) used for tracking metrics.

3.1.29. .status.operationHistory.deprovision

Description

OperationMetric contains metadata about an operation (inspection, provisioning, etc.) used for tracking metrics.

Type

object

Property	Type	Description
end	^^	
start	^^	

3.1.30. .status.operationHistory.inspect

Description

OperationMetric contains metadata about an operation (inspection, provisioning, etc.) used for tracking metrics.

Type

object

Property	Type	Description
end	^^	
start	^^	

3.1.31. .status.operationHistory.provision

Description

OperationMetric contains metadata about an operation (inspection, provisioning, etc.) used for tracking metrics.

Type

object

Property	Type	Description
end	..	
start	..	

3.1.32. .status.operationHistory.register

Description

OperationMetric contains metadata about an operation (inspection, provisioning, etc.) used for tracking metrics.

Type

object

Property	Type	Description
end	..	
start	..	

3.1.33. .status.provisioning

Description

Information tracked by the provisioner.

Type

object

Required

- **ID**
- **state**

Property	Type	Description
ID	string	The hosts's ID from the underlying provisioning tool (e.g. the Ironic node UUID).
bootMode	string	BootMode indicates the boot mode used to provision the host.
customDeploy	object	Custom deploy procedure applied to the host.

Property	Type	Description
firmware	object	The firmware settings that have been applied.
image	object	Image holds the details of the last image successfully provisioned to the host.
raid	object	The RAID configuration that has been applied.
rootDeviceHints	object	The root device hints used to provision the host.
state	string	An indicator for what the provisioner is doing with the host.

3.1.34. .status.provisioning.customDeploy

Description

Custom deploy procedure applied to the host.

Type

object

Required

- **method**

Property	Type	Description
method	string	Custom deploy method name. This name is specific to the deploy ramdisk used. If you don't have a custom deploy ramdisk, you shouldn't use CustomDeploy.

3.1.35. .status.provisioning.firmware

Description

The firmware settings that have been applied.

Type

object

Property	Type	Description
simultaneousMultithreadingEnabled	boolean	Allows a single physical processor core to appear as several logical processors.
sriovEnabled	boolean	SR-IOV support enables a hypervisor to create virtual instances of a PCI-express device, potentially increasing performance.
virtualizationEnabled	boolean	Supports the virtualization of platform hardware.

3.1.36. .status.provisioning.image

Description

Image holds the details of the last image successfully provisioned to the host.

Type

object

Required

- **url**

Property	Type	Description
checksum	string	Checksum is the checksum for the image. Required for all formats except for "live-iso".
checksumType	string	ChecksumType is the checksum algorithm for the image, e.g md5, sha256 or sha512. The special value "auto" can be used to detect the algorithm from the checksum. If missing, MD5 is used. If in doubt, use "auto".
format	string	Format contains the format of the image (raw, qcow2, ...). When set to "live-iso", an ISO 9660 image referenced by the url will be live-booted and not deployed to disk.

Property	Type	Description
url	string	URL is a location of an image to deploy.

3.1.37. .status.provisioning.raid

Description

The RAID configuration that has been applied.

Type

object

Property	Type	Description
hardwareRAIDVolumes	^^	The list of logical disks for hardware RAID, if rootDeviceHints isn't used, first volume is root volume. You can set the value of this field to [] to clear all the hardware RAID configurations.
softwareRAIDVolumes	^^	The list of logical disks for software RAID, if rootDeviceHints isn't used, first volume is root volume. If HardwareRAIDVolumes is set this item will be invalid. The number of created Software RAID devices must be 1 or 2. If there is only one Software RAID device, it has to be a RAID-1. If there are two, the first one has to be a RAID-1, while the RAID level for the second one can be 0, 1, or 1+0. As the first RAID device will be the deployment device, enforcing a RAID-1 reduces the risk of ending up with a non-booting host in case of a disk failure. Software RAID will always be deleted.

3.1.38. .status.provisioning.rootDeviceHints

Description

The root device hints used to provision the host.

Type

object

Property	Type	Description
deviceName	string	A Linux device name like <code>"/dev/vda"</code> , or a by-path link to it like <code>"/dev/disk/by-path/pci-0000:01:00.0-scsi-0:2:0:0"</code> . The hint must match the actual value exactly.
hctl	string	A SCSI bus address like <code>0:0:0:0</code> . The hint must match the actual value exactly.
minSizeGigabytes	integer	The minimum size of the device in Gigabytes.
model	string	A vendor-specific device identifier. The hint can be a substring of the actual value.
rotational	boolean	True if the device should use spinning media, false otherwise.
serialNumber	string	Device serial number. The hint must match the actual value exactly.
vendor	string	The name of the vendor or manufacturer of the device. The hint can be a substring of the actual value.
wwn	string	Unique storage identifier. The hint must match the actual value exactly.
wwnVendorExtension	string	Unique vendor storage identifier. The hint must match the actual value exactly.
wwnWithExtension	string	Unique storage identifier with the vendor extension appended. The hint must match the actual value exactly.

3.1.39. `.status.triedCredentials`

Description

The last credentials we sent to the provisioning backend.

Type

object

Property	Type	Description
credentials	object	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
credentialsVersion	string	

3.1.40. .status.triedCredentials.credentials**Description**

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

Type**object**

Property	Type	Description
name	string	name is unique within a namespace to reference a secret resource.
namespace	string	namespace defines the space within which the secret name must be unique.

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/metal3.io/v1alpha1/baremetalhosts**
 - **GET**: list objects of kind BareMetalHost
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts**
 - **DELETE**: delete collection of BareMetalHost
 - **GET**: list objects of kind BareMetalHost
 - **POST**: create a BareMetalHost
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts/{name}**
 - **DELETE**: delete a BareMetalHost
 - **GET**: read the specified BareMetalHost

- **PATCH**: partially update the specified BareMetalHost
- **PUT**: replace the specified BareMetalHost
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts/{name}/status**
 - **GET**: read status of the specified BareMetalHost
 - **PATCH**: partially update status of the specified BareMetalHost
 - **PUT**: replace status of the specified BareMetalHost

3.2.1. /apis/metal3.io/v1alpha1/baremetalhosts

HTTP method

GET

Description

list objects of kind BareMetalHost

Table 3.1. HTTP responses

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

3.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts

HTTP method

DELETE

Description

delete collection of BareMetalHost

Table 3.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind BareMetalHost

Table 3.3. HTTP responses

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

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

create a BareMetalHost

Table 3.4. Query parameters

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

Table 3.5. Body parameters

Parameter	Type	Description
body	BareMetalHost schema	

Table 3.6. HTTP responses

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

3.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts/{name}

Table 3.7. Global path parameters

Parameter	Type	Description
name	string	name of the BareMetalHost

HTTP method

DELETE

Description

delete a BareMetalHost

Table 3.8. Query parameters

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

Table 3.9. HTTP responses

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

HTTP method

GET

Description

read the specified BareMetalHost

Table 3.10. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified BareMetalHost

Table 3.11. Query parameters

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

Table 3.12. HTTP responses

HTTP code	Response body
200 - OK	BareMetalHost schema

HTTP code	Response body
401 - Unauthorized	Empty

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

replace the specified BareMetalHost

Table 3.13. Query parameters

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

Table 3.14. Body parameters

Parameter	Type	Description
body	BareMetalHost schema	

Table 3.15. HTTP responses

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

3.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/baremetalhosts/{name}/s

Table 3.16. Global path parameters

Parameter	Type	Description
name	string	name of the BareMetalHost

HTTP method

GET

Description

read status of the specified BareMetalHost

Table 3.17. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified BareMetalHost

Table 3.18. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <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 3.19. HTTP responses

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

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

replace status of the specified BareMetalHost

Table 3.20. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <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 3.21. Body parameters

Parameter	Type	Description
body	BareMetalHost schema	

Table 3.22. HTTP responses

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

CHAPTER 4. DATAIMAGE [METAL3.IO/V1ALPHA1]

Description

Datamodel is the Schema for the dataimages API.

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	DatamodelSpec defines the desired state of Datamodel.
status	object	DatamodelStatus defines the observed state of Datamodel.

4.1.1. .spec

Description

DataltImageSpec defines the desired state of DataltImage.

Type

object

Required

- **url**

Property	Type	Description
url	string	Url is the address of the dataltImage that we want to attach to a BareMetalHost

4.1.2. .status

Description

DataltImageStatus defines the observed state of DataltImage.

Type

object

Property	Type	Description
attachedImage	object	Currently attached DataltImage
error	object	Error count and message when attaching/detaching
lastReconciled	string	Time of last reconciliation

4.1.3. .status.attachedImage

Description

Currently attached DataltImage

Type

object

Required

- **url**

Property	Type	Description
url	string	

4.1.4. .status.error

Description

Error count and message when attaching/detaching

Type

object

Required

- **count**
- **message**

Property	Type	Description
count	integer	
message	string	

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/metal3.io/v1alpha1/dataimages**
 - **GET**: list objects of kind DataImage
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages**
 - **DELETE**: delete collection of DataImage
 - **GET**: list objects of kind DataImage
 - **POST**: create a DataImage
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages/{name}**
 - **DELETE**: delete a DataImage
 - **GET**: read the specified DataImage
 - **PATCH**: partially update the specified DataImage
 - **PUT**: replace the specified DataImage
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages/{name}/status**
 - **GET**: read status of the specified DataImage
 - **PATCH**: partially update status of the specified DataImage
 - **PUT**: replace status of the specified DataImage

4.2.1. /apis/metal3.io/v1alpha1/dataimages

HTTP method

GET**Description**

list objects of kind DataImage

Table 4.1. HTTP responses

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

4.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages**HTTP method****DELETE****Description**

delete collection of DataImage

Table 4.2. HTTP responses

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

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

list objects of kind DataImage

Table 4.3. HTTP responses

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

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

create a DataImage

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	DatalImage schema	

Table 4.6. HTTP responses

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

4.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages/{name}

Table 4.7. Global path parameters

Parameter	Type	Description
name	string	name of the DataImage

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

delete a DataImage

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 DataImage

Table 4.10. HTTP responses

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

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

partially update the specified DataImage

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	Datalmage schema
401 - Unauthorized	Empty

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

replace the specified Datalmage

Table 4.13. Query parameters

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

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

Table 4.14. Body parameters

Parameter	Type	Description
body	Datamodel schema	

Table 4.15. HTTP responses

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

4.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/dataimages/{name}/statu

Table 4.16. Global path parameters

Parameter	Type	Description
name	string	name of the DataImage

HTTP method

GET

Description

read status of the specified DataImage

Table 4.17. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified DataImage

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: <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 4.19. HTTP responses

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

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

replace status of the specified DatImage

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: <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 4.21. Body parameters

Parameter	Type	Description
body	Datalmage schema	

Table 4.22. HTTP responses

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

CHAPTER 5. FIRMWARESCHEMA [METAL3.IO/V1ALPHA1]

Description

FirmwareSchema is the Schema for the firmwareschemas API.

Type

object

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	FirmwareSchemaSpec defines the desired state of FirmwareSchema.

5.1.1. .spec

Description

FirmwareSchemaSpec defines the desired state of FirmwareSchema.

Type

object

Required

- **schema**

Property	Type	Description
hardwareModel	string	The hardware model associated with this schema
hardwareVendor	string	The hardware vendor associated with this schema
schema	object	Map of firmware name to schema
schema{}	object	Additional data describing the firmware setting.

5.1.2. .spec.schema**Description**

Map of firmware name to schema

Type

object

5.1.3. .spec.schema{}**Description**

Additional data describing the firmware setting.

Type

object

Property	Type	Description
allowable_values	array (string)	The allowable value for an Enumeration type setting.
attribute_type	string	The type of setting.
lower_bound	integer	The lowest value for an Integer type setting.
max_length	integer	Maximum length for a String type setting.
min_length	integer	Minimum length for a String type setting.

Property	Type	Description
read_only	boolean	Whether or not this setting is read only.
unique	boolean	Whether or not this setting's value is unique to this node, e.g. a serial number.
upper_bound	integer	The highest value for an Integer type setting.

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/metal3.io/v1alpha1/firmwareschemas**
 - **GET**: list objects of kind FirmwareSchema
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/firmwareschemas**
 - **DELETE**: delete collection of FirmwareSchema
 - **GET**: list objects of kind FirmwareSchema
 - **POST**: create a FirmwareSchema
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/firmwareschemas/{name}**
 - **DELETE**: delete a FirmwareSchema
 - **GET**: read the specified FirmwareSchema
 - **PATCH**: partially update the specified FirmwareSchema
 - **PUT**: replace the specified FirmwareSchema

5.2.1. /apis/metal3.io/v1alpha1/firmwareschemas

HTTP method

GET

Description

list objects of kind FirmwareSchema

Table 5.1. HTTP responses

HTTP code	Reponse body
200 - OK	FirmwareSchemaList schema

HTTP code	Reponse body
401 - Unauthorized	Empty

5.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/firmwareschemas

HTTP method

DELETE

Description

delete collection of FirmwareSchema

Table 5.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind FirmwareSchema

Table 5.3. HTTP responses

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

HTTP method

POST

Description

create a FirmwareSchema

Table 5.4. 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: <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 5.5. Body parameters

Parameter	Type	Description
body	FirmwareSchema schema	

Table 5.6. HTTP responses

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

5.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/firmwareschemas/{name}

Table 5.7. Global path parameters

Parameter	Type	Description
name	string	name of the FirmwareSchema

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

delete a FirmwareSchema

Table 5.8. 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 5.9. HTTP responses

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

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

read the specified FirmwareSchema

Table 5.10. HTTP responses

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

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

partially update the specified FirmwareSchema

Table 5.11. 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 5.12. HTTP responses

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

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

replace the specified FirmwareSchema

Table 5.13. 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 5.14. Body parameters

Parameter	Type	Description
body	FirmwareSchema schema	

Table 5.15. HTTP responses

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

CHAPTER 6. HARDWAREDATA [METAL3.IO/V1ALPHA1]

Description

HardwareData is the Schema for the hardwaredata API.

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	HardwareDataSpec defines the desired state of HardwareData.

6.1.1. .spec

Description

HardwareDataSpec defines the desired state of HardwareData.

Type

object

Property	Type	Description
hardware	object	The hardware discovered on the host during its inspection.

6.1.2. .spec.hardware

Description

The hardware discovered on the host during its inspection.

Type

object

Property	Type	Description
cpu	object	Details of the CPU(s) in the system.
firmware	object	System firmware information.
hostname	string	
nics	array	List of network interfaces for the host.
nics[]	object	NIC describes one network interface on the host.
ramMebibytes	integer	The host's amount of memory in Mebibytes.
storage	array	List of storage (disk, SSD, etc.) available to the host.
storage[]	object	Storage describes one storage device (disk, SSD, etc.) on the host.
systemVendor	object	System vendor information.

6.1.3. .spec.hardware.cpu

Description

Details of the CPU(s) in the system.

Type

object

Property	Type	Description
arch	string	
clockMegahertz	number	ClockSpeed is a clock speed in MHz
count	integer	
flags	array (string)	
model	string	

6.1.4. .spec.hardware.firmware

Description

System firmware information.

Type

object

Property	Type	Description
bios	object	The BIOS for this firmware

6.1.5. .spec.hardware.firmware.bios

Description

The BIOS for this firmware

Type

object

Property	Type	Description
date	string	The release/build date for this BIOS
vendor	string	The vendor name for this BIOS
version	string	The version of the BIOS

6.1.6. .spec.hardware.nics

Description

List of network interfaces for the host.

Type

array

6.1.7. .spec.hardware.nics[]

Description

NIC describes one network interface on the host.

Type

object

Property	Type	Description
ip	string	The IP address of the interface. This will be an IPv4 or IPv6 address if one is present. If both IPv4 and IPv6 addresses are present in a dual-stack environment, two nics will be output, one with each IP.
mac	string	The device MAC address
model	string	The vendor and product IDs of the NIC, e.g. "0x8086 0x1572"
name	string	The name of the network interface, e.g. "en0"
pxe	boolean	Whether the NIC is PXE Bootable
speedGbps	integer	The speed of the device in Gigabits per second
vlanId	integer	The untagged VLAN ID
vlangs	array	The VLANs available
vlangs[]	object	VLAN represents the name and ID of a VLAN.

6.1.8. .spec.hardware.nics[].vlangs

Description

The VLANs available

Type

array

6.1.9. .spec.hardware.nics[].vlangs[]

Description

VLAN represents the name and ID of a VLAN.

Type

object

Property	Type	Description
id	integer	VLANID is a 12-bit 802.1Q VLAN identifier
name	string	

6.1.10. .spec.hardware.storage**Description**

List of storage (disk, SSD, etc.) available to the host.

Type

array

6.1.11. .spec.hardware.storage[]**Description**

Storage describes one storage device (disk, SSD, etc.) on the host.

Type

object

Property	Type	Description
alternateNames	array (string)	A list of alternate Linux device names of the disk, e.g. <code>"/dev/sda"</code> . Note that this list is not exhaustive, and names may not be stable across reboots.
hctl	string	The SCSI location of the device
model	string	Hardware model
name	string	A Linux device name of the disk, e.g. <code>"/dev/disk/by-path/pci-0000:01:00.0-scsi-0:2:0:0"</code> . This will be a name that is stable across reboots if one is available.

Property	Type	Description
rotational	boolean	Whether this disk represents rotational storage. This field is not recommended for usage, please prefer using 'Type' field instead, this field will be deprecated eventually.
serialNumber	string	The serial number of the device
sizeBytes	integer	The size of the disk in Bytes
type	string	Device type, one of: HDD, SSD, NVME.
vendor	string	The name of the vendor of the device
wwn	string	The WWN of the device
wwnVendorExtension	string	The WWN Vendor extension of the device
wwnWithExtension	string	The WWN with the extension

6.1.12. .spec.hardware.systemVendor

Description

System vendor information.

Type

object

Property	Type	Description
manufacturer	string	
productName	string	
serialNumber	string	

6.2. API ENDPOINTS

The following API endpoints are available:

- `/apis/metal3.io/v1alpha1/hardwaredata`

- **GET**: list objects of kind HardwareData
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/hardwaredata**
 - **DELETE**: delete collection of HardwareData
 - **GET**: list objects of kind HardwareData
 - **POST**: create a HardwareData
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/hardwaredata/{name}**
 - **DELETE**: delete a HardwareData
 - **GET**: read the specified HardwareData
 - **PATCH**: partially update the specified HardwareData
 - **PUT**: replace the specified HardwareData

6.2.1. /apis/metal3.io/v1alpha1/hardwaredata

HTTP method

GET

Description

list objects of kind HardwareData

Table 6.1. HTTP responses

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

6.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hardwaredata

HTTP method

DELETE

Description

delete collection of HardwareData

Table 6.2. HTTP responses

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

HTTP method

GET**Description**

list objects of kind HardwareData

Table 6.3. HTTP responses

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

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

create a HardwareData

Table 6.4. 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 6.5. Body parameters

Parameter	Type	Description
body	HardwareData schema	

Table 6.6. HTTP responses

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

6.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hardwaredata/{name}

Table 6.7. Global path parameters

Parameter	Type	Description
name	string	name of the HardwareData

HTTP method

DELETE

Description

delete a HardwareData

Table 6.8. 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 6.9. 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 HardwareData

Table 6.10. HTTP responses

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

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

partially update the specified HardwareData

Table 6.11. 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

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 6.12. HTTP responses

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

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

replace the specified HardwareData

Table 6.13. 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: <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 6.14. Body parameters

Parameter	Type	Description
body	HardwareData schema	

Table 6.15. HTTP responses

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

CHAPTER 7. HOSTFIRMWARECOMPONENTS [METAL3.IO/V1ALPHA1]

Description

HostFirmwareComponents is the Schema for the hostfirmwarecomponents API.

Type

object

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	HostFirmwareComponentsSpec defines the desired state of HostFirmwareComponents.
status	object	HostFirmwareComponentsStatus defines the observed state of HostFirmwareComponents.

7.1.1. .spec

Description

HostFirmwareComponentsSpec defines the desired state of HostFirmwareComponents.

Type

object

Required

- **updates**

Property	Type	Description
updates	array	
updates[]	object	FirmwareUpdate defines a firmware update specification.

7.1.2. .spec.updates

Description

Type

array

7.1.3. .spec.updates[]

Description

FirmwareUpdate defines a firmware update specification.

Type

object

Required

- **component**
- **url**

Property	Type	Description
component	string	
url	string	

7.1.4. .status

Description

HostFirmwareComponentsStatus defines the observed state of HostFirmwareComponents.

Type
object

Property	Type	Description
components	array	Components is the list of all available firmware components and their information.
components[]	object	FirmwareComponentStatus defines the status of a firmware component.
conditions	array	Track whether updates stored in the spec are valid based on the schema
conditions[]	object	Condition contains details for one aspect of the current state of this API Resource.
lastUpdated	string	Time that the status was last updated
updates	array	Updates is the list of all firmware components that should be updated they are specified via name and url fields.
updates[]	object	FirmwareUpdate defines a firmware update specification.

7.1.5. .status.components

Description

Components is the list of all available firmware components and their information.

Type
array

7.1.6. .status.components[]

Description

FirmwareComponentStatus defines the status of a firmware component.

Type
object

Required

- **component**

- **initialVersion**

Property	Type	Description
component	string	
currentVersion	string	
initialVersion	string	
lastVersionFlashed	string	
updatedAt	string	

7.1.7. .status.conditions

Description

Track whether updates stored in the spec are valid based on the schema

Type

array

7.1.8. .status.conditions[]

Description

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

Type

object

Required

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

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

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

7.1.9. .status.updates

Description

Updates is the list of all firmware components that should be updated they are specified via name and url fields.

Type

array

7.1.10. .status.updates[]

Description

FirmwareUpdate defines a firmware update specification.

Type

object

Required

- **component**
- **url**

Property	Type	Description
component	string	
url	string	

7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/metal3.io/v1alpha1/hostfirmwarecomponents**
 - **GET**: list objects of kind HostFirmwareComponents
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents**
 - **DELETE**: delete collection of HostFirmwareComponents
 - **GET**: list objects of kind HostFirmwareComponents
 - **POST**: create HostFirmwareComponents
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents/{name}**
 - **DELETE**: delete HostFirmwareComponents
 - **GET**: read the specified HostFirmwareComponents
 - **PATCH**: partially update the specified HostFirmwareComponents
 - **PUT**: replace the specified HostFirmwareComponents
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents/{name}/status**
 - **GET**: read status of the specified HostFirmwareComponents
 - **PATCH**: partially update status of the specified HostFirmwareComponents
 - **PUT**: replace status of the specified HostFirmwareComponents

7.2.1. /apis/metal3.io/v1alpha1/hostfirmwarecomponents

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

list objects of kind HostFirmwareComponents

Table 7.1. HTTP responses

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

7.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents**HTTP method****DELETE****Description**

delete collection of HostFirmwareComponents

Table 7.2. HTTP responses

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

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

list objects of kind HostFirmwareComponents

Table 7.3. HTTP responses

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

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

create HostFirmwareComponents

Table 7.4. Query parameters

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

Table 7.5. Body parameters

Parameter	Type	Description
body	HostFirmwareComponents schema	

Table 7.6. HTTP responses

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

7.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents/

Table 7.7. Global path parameters

Parameter	Type	Description
name	string	name of the HostFirmwareComponents

HTTP method

DELETE

Description

delete HostFirmwareComponents

Table 7.8. Query parameters

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

Table 7.9. HTTP responses

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

HTTP method

GET

Description

read the specified HostFirmwareComponents

Table 7.10. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified HostFirmwareComponents

Table 7.11. Query parameters

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

Table 7.12. HTTP responses

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

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

replace the specified HostFirmwareComponents

Table 7.13. Query parameters

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

Table 7.14. Body parameters

Parameter	Type	Description
body	HostFirmwareComponents schema	

Table 7.15. HTTP responses

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

7.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwarecomponents/

Table 7.16. Global path parameters

Parameter	Type	Description
name	string	name of the HostFirmwareComponents

HTTP method

GET

Description

read status of the specified HostFirmwareComponents

Table 7.17. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified HostFirmwareComponents

Table 7.18. Query parameters

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

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <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 7.19. HTTP responses

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

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

replace status of the specified HostFirmwareComponents

Table 7.20. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <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 7.21. Body parameters

Parameter	Type	Description
body	HostFirmwareComponents schema	

Table 7.22. HTTP responses

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

CHAPTER 8. HOSTFIRMWARESETTINGS [METAL3.IO/V1ALPHA1]

Description

HostFirmwareSettings is the Schema for the hostfirmwaresettings API.

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	HostFirmwareSettingsSpec defines the desired state of HostFirmwareSettings.
status	object	HostFirmwareSettingsStatus defines the observed state of HostFirmwareSettings.

8.1.1. .spec

Description

HostFirmwareSettingsSpec defines the desired state of HostFirmwareSettings.

Type

object

Required

- **settings**

Property	Type	Description
settings	integer-or-string	Settings are the desired firmware settings stored as name/value pairs.

8.1.2. .status

Description

HostFirmwareSettingsStatus defines the observed state of HostFirmwareSettings.

Type

object

Required

- **settings**

Property	Type	Description
conditions	array	Track whether settings stored in the spec are valid based on the schema
conditions[]	object	Condition contains details for one aspect of the current state of this API Resource.
lastUpdated	string	Time that the status was last updated
schema	object	FirmwareSchema is a reference to the Schema used to describe each FirmwareSetting. By default, this will be a Schema in the same Namespace as the settings but it can be overwritten in the Spec

Property	Type	Description
settings	object (string)	Settings are the firmware settings stored as name/value pairs

8.1.3. .status.conditions

Description

Track whether settings stored in the spec are valid based on the schema

Type

array

8.1.4. .status.conditions[]

Description

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

Type

object

Required

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

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

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

8.1.5. .status.schema

Description

FirmwareSchema is a reference to the Schema used to describe each FirmwareSetting. By default, this will be a Schema in the same Namespace as the settings but it can be overwritten in the Spec

Type

object

Required

- **name**
- **namespace**

Property	Type	Description
name	string	name is the reference to the schema.
namespace	string	namespace is the namespace of the where the schema is stored.

8.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/metal3.io/v1alpha1/hostfirmwaresettings**
 - **GET**: list objects of kind HostFirmwareSettings
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings**
 - **DELETE**: delete collection of HostFirmwareSettings
 - **GET**: list objects of kind HostFirmwareSettings
 - **POST**: create HostFirmwareSettings
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings/{name}**
 - **DELETE**: delete HostFirmwareSettings
 - **GET**: read the specified HostFirmwareSettings
 - **PATCH**: partially update the specified HostFirmwareSettings
 - **PUT**: replace the specified HostFirmwareSettings
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings/{name}/status**
 - **GET**: read status of the specified HostFirmwareSettings
 - **PATCH**: partially update status of the specified HostFirmwareSettings
 - **PUT**: replace status of the specified HostFirmwareSettings

8.2.1. /apis/metal3.io/v1alpha1/hostfirmwaresettings

HTTP method

GET

Description

list objects of kind HostFirmwareSettings

Table 8.1. HTTP responses

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

8.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings

HTTP method

DELETE

Description

delete collection of HostFirmwareSettings

Table 8.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind HostFirmwareSettings

Table 8.3. HTTP responses

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

HTTP method

POST

Description

create HostFirmwareSettings

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

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 8.5. Body parameters

Parameter	Type	Description
body	HostFirmwareSetting s schema	

Table 8.6. HTTP responses

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

8.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings/{name}

Table 8.7. Global path parameters

Parameter	Type	Description
name	string	name of the HostFirmwareSettings

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

delete HostFirmwareSettings

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 HostFirmwareSettings

Table 8.10. HTTP responses

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

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

partially update the specified HostFirmwareSettings

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	HostFirmwareSettings schema
401 - Unauthorized	Empty

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

replace the specified HostFirmwareSettings

Table 8.13. 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 8.14. Body parameters

Parameter	Type	Description
body	HostFirmwareSetting s schema	

Table 8.15. HTTP responses

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

8.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/hostfirmwaresettings/{name}

Table 8.16. Global path parameters

Parameter	Type	Description
name	string	name of the HostFirmwareSettings

HTTP method

GET

Description

read status of the specified HostFirmwareSettings

Table 8.17. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified HostFirmwareSettings

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	HostFirmwareSettings schema
401 - Unauthorized	Empty

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

replace status of the specified HostFirmwareSettings

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	HostFirmwareSetting s schema	

Table 8.22. HTTP responses

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

CHAPTER 9. METAL3REMEDICATION

[INFRASTRUCTURE.CLUSTER.X-K8S.IO/V1BETA1]

Description

Metal3Remediation is the Schema for the metal3remediations API.

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	Metal3RemediationSpec defines the desired state of Metal3Remediation.
status	object	Metal3RemediationStatus defines the observed state of Metal3Remediation.

9.1.1. .spec

Description

Metal3RemediationSpec defines the desired state of Metal3Remediation.

Type

object

Property	Type	Description
strategy	object	Strategy field defines remediation strategy.

9.1.2. .spec.strategy

Description

Strategy field defines remediation strategy.

Type

object

Property	Type	Description
retryLimit	integer	Sets maximum number of remediation retries.
timeout	string	Sets the timeout between remediation retries.
type	string	Type of remediation.

9.1.3. .status

Description

Metal3RemediationStatus defines the observed state of Metal3Remediation.

Type

object

Property	Type	Description
lastRemediated	string	LastRemediated identifies when the host was last remediated
phase	string	Phase represents the current phase of machine remediation. E.g. Pending, Running, Done etc.

Property	Type	Description
retryCount	integer	RetryCount can be used as a counter during the remediation. Field can hold number of reboots etc.

9.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/infrastructure.cluster.x-k8s.io/v1beta1/metal3remediations**
 - **GET**: list objects of kind Metal3Remediation
- **/apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations**
 - **DELETE**: delete collection of Metal3Remediation
 - **GET**: list objects of kind Metal3Remediation
 - **POST**: create a Metal3Remediation
- **/apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations/{name}**
 - **DELETE**: delete a Metal3Remediation
 - **GET**: read the specified Metal3Remediation
 - **PATCH**: partially update the specified Metal3Remediation
 - **PUT**: replace the specified Metal3Remediation
- **/apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations/{name}/status**
 - **GET**: read status of the specified Metal3Remediation
 - **PATCH**: partially update status of the specified Metal3Remediation
 - **PUT**: replace status of the specified Metal3Remediation

9.2.1. /apis/infrastructure.cluster.x-k8s.io/v1beta1/metal3remediations

HTTP method

GET

Description

list objects of kind Metal3Remediation

Table 9.1. HTTP responses

HTTP code	Reponse body
200 - OK	Metal3RemediationList schema
401 - Unauthorized	Empty

9.2.2. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations

HTTP method

DELETE

Description

delete collection of Metal3Remediation

Table 9.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind Metal3Remediation

Table 9.3. HTTP responses

HTTP code	Reponse body
200 - OK	Metal3RemediationList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Metal3Remediation

Table 9.4. 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 9.5. Body parameters

Parameter	Type	Description
body	Metal3Remediation schema	

Table 9.6. HTTP responses

HTTP code	Response body
200 - OK	Metal3Remediation schema
201 - Created	Metal3Remediation schema
202 - Accepted	Metal3Remediation schema
401 - Unauthorized	Empty

9.2.3. `/apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations/{name}`

Table 9.7. Global path parameters

Parameter	Type	Description
name	string	name of the Metal3Remediation

HTTP method

DELETE

Description

delete a Metal3Remediation

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

HTTP method

GET

Description

read the specified Metal3Remediation

Table 9.10. HTTP responses

HTTP code	Reponse body
200 - OK	Metal3Remediation schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Metal3Remediation

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	Metal3Remediation schema
401 - Unauthorized	Empty

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

replace the specified Metal3Remediation

Table 9.13. 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 9.14. Body parameters

Parameter	Type	Description
body	Metal3Remediation schema	

Table 9.15. HTTP responses

HTTP code	Response body
200 - OK	Metal3Remediation schema
201 - Created	Metal3Remediation schema
401 - Unauthorized	Empty

9.2.4. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediations/{name}/status

Table 9.16. Global path parameters

Parameter	Type	Description
name	string	name of the Metal3Remediation

HTTP method

GET

Description

read status of the specified Metal3Remediation

Table 9.17. HTTP responses

HTTP code	Reponse body
200 - OK	Metal3Remediation schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Metal3Remediation

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: - 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	Metal3Remediation schema
401 - Unauthorized	Empty

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

replace status of the specified Metal3Remediation

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: - 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	Metal3Remediation schema	

Table 9.22. HTTP responses

HTTP code	Response body
200 - OK	Metal3Remediation schema
201 - Created	Metal3Remediation schema
401 - Unauthorized	Empty

CHAPTER 10. METAL3REMIEDIATIONTEMPLATE [INFRASTRUCTURE.CLUSTER.X-K8S.IO/V1BETA1]

Description

Metal3RemediationTemplate is the Schema for the metal3remediationtemplates API.

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	Metal3RemediationTemplateSpec defines the desired state of Metal3RemediationTemplate.
status	object	Metal3RemediationTemplateStatus defines the observed state of Metal3RemediationTemplate.

10.1.1. .spec

Description

Metal3RemediationTemplateSpec defines the desired state of Metal3RemediationTemplate.

Type

object

Required

- **template**

Property	Type	Description
template	object	Metal3RemediationTemplateResource describes the data needed to create a Metal3Remediation from a template.

10.1.2. .spec.template

Description

Metal3RemediationTemplateResource describes the data needed to create a Metal3Remediation from a template.

Type

object

Required

- **spec**

Property	Type	Description
spec	object	Spec is the specification of the desired behavior of the Metal3Remediation.

10.1.3. .spec.template.spec

Description

Spec is the specification of the desired behavior of the Metal3Remediation.

Type

object

Property	Type	Description
strategy	object	Strategy field defines remediation strategy.

10.1.4. .spec.template.spec.strategy

Description

Strategy field defines remediation strategy.

Type

object

Property	Type	Description
retryLimit	integer	Sets maximum number of remediation retries.
timeout	string	Sets the timeout between remediation retries.
type	string	Type of remediation.

10.1.5. .status

Description

Metal3RemediationTemplateStatus defines the observed state of Metal3RemediationTemplate.

Type

object

Required

- **status**

Property	Type	Description
status	object	Metal3RemediationStatus defines the observed state of Metal3Remediation

10.1.6. .status.status

Description

Metal3RemediationStatus defines the observed state of Metal3Remediation

Type

object

Property	Type	Description
lastRemediated	string	LastRemediated identifies when the host was last remediated

Property	Type	Description
phase	string	Phase represents the current phase of machine remediation. E.g. Pending, Running, Done etc.
retryCount	integer	RetryCount can be used as a counter during the remediation. Field can hold number of reboots etc.

10.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/infrastructure.cluster.x-k8s.io/v1beta1/metal3remediationtemplates**
 - **GET**: list objects of kind Metal3RemediationTemplate
- **/apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates**
 - **DELETE**: delete collection of Metal3RemediationTemplate
 - **GET**: list objects of kind Metal3RemediationTemplate
 - **POST**: create a Metal3RemediationTemplate
- **/apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates/{name}**
 - **DELETE**: delete a Metal3RemediationTemplate
 - **GET**: read the specified Metal3RemediationTemplate
 - **PATCH**: partially update the specified Metal3RemediationTemplate
 - **PUT**: replace the specified Metal3RemediationTemplate
- **/apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates/{name}/status**
 - **GET**: read status of the specified Metal3RemediationTemplate
 - **PATCH**: partially update status of the specified Metal3RemediationTemplate
 - **PUT**: replace status of the specified Metal3RemediationTemplate

10.2.1. /apis/infrastructure.cluster.x-k8s.io/v1beta1/metal3remediationtemplates

HTTP method

GET

Description

list objects of kind Metal3RemediationTemplate

Table 10.1. HTTP responses

HTTP code	Reponse body
200 - OK	Metal3RemediationTemplateList schema
401 - Unauthorized	Empty

10.2.2. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates

HTTP method

DELETE

Description

delete collection of Metal3RemediationTemplate

Table 10.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind Metal3RemediationTemplate

Table 10.3. HTTP responses

HTTP code	Reponse body
200 - OK	Metal3RemediationTemplateList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Metal3RemediationTemplate

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	Metal3RemediationTemplate schema	

Table 10.6. HTTP responses

HTTP code	Response body
200 - OK	Metal3RemediationTemplate schema
201 - Created	Metal3RemediationTemplate schema
202 - Accepted	Metal3RemediationTemplate schema
401 - Unauthorized	Empty

10.2.3. `/apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates/{name}`

Table 10.7. Global path parameters

Parameter	Type	Description
name	string	name of the Metal3RemediationTemplate

HTTP method

DELETE

Description

delete a Metal3RemediationTemplate

Table 10.8. Query parameters

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

Table 10.9. HTTP responses

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

HTTP method

GET

Description

read the specified Metal3RemediationTemplate

Table 10.10. HTTP responses

HTTP code	Reponse body
200 - OK	Metal3RemediationTemplate schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Metal3RemediationTemplate

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	Metal3RemediationTemplate schema
401 - Unauthorized	Empty

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

replace the specified Metal3RemediationTemplate

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	Metal3RemediationTemplate schema	

Table 10.15. HTTP responses

HTTP code	Response body
200 - OK	Metal3RemediationTemplate schema
201 - Created	Metal3RemediationTemplate schema
401 - Unauthorized	Empty

10.2.4. /apis/infrastructure.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/metal3remediationtemplates/{name}/stat

Table 10.16. Global path parameters

Parameter	Type	Description
name	string	name of the Metal3RemediationTemplate

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

read status of the specified Metal3RemediationTemplate

Table 10.17. HTTP responses

HTTP code	Reponse body
200 - OK	Metal3RemediationTemplate schema
401 - Unauthorized	Empty

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

partially update status of the specified Metal3RemediationTemplate

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	Metal3RemediationTemplate schema
401 - Unauthorized	Empty

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

replace status of the specified Metal3RemediationTemplate

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	Metal3RemediationTemplate schema	

Table 10.22. HTTP responses

HTTP code	Response body
200 - OK	Metal3RemediationTemplate schema
201 - Created	Metal3RemediationTemplate schema
401 - Unauthorized	Empty

CHAPTER 11. PREPROVISIONINGIMAGE [METAL3.IO/V1ALPHA1]

Description

PreprovisioningImage is the Schema for the preprovisioningimages API.

Type

object

11.1. 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	PreprovisioningImageSpec defines the desired state of PreprovisioningImage.
status	object	PreprovisioningImageStatus defines the observed state of PreprovisioningImage.

11.1.1. .spec

Description

PreprovisioningImageSpec defines the desired state of PreprovisioningImage.

Type

object

Property	Type	Description
acceptFormats	array (string)	acceptFormats is a list of acceptable image formats.
architecture	string	architecture is the processor architecture for which to build the image.
networkDataName	string	networkDataName is the name of a Secret in the local namespace that contains network data to build in to the image.

11.1.2. .status

Description

PreprovisioningImageStatus defines the observed state of PreprovisioningImage.

Type

object

Property	Type	Description
architecture	string	architecture is the processor architecture for which the image is built
conditions	array	conditions describe the state of the built image
conditions[]	object	Condition contains details for one aspect of the current state of this API Resource.
extraKernelParams	string	extraKernelParams is a string with extra parameters to pass to the kernel when booting the image over network. Only makes sense for initrd images.

Property	Type	Description
format	string	format is the type of image that is available at the download url: either iso or initrd.
imageUrl	string	imageUrl is the URL from which the built image can be downloaded.
kernelUrl	string	kernelUrl is the URL from which the kernel of the image can be downloaded. Only makes sense for initrd images.
networkData	object	networkData is a reference to the version of the Secret containing the network data used to build the image.

11.1.3. .status.conditions

Description

conditions describe the state of the built image

Type

array

11.1.4. .status.conditions[]

Description

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

Type

object

Required

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

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

11.1.5. .status.networkData

Description

networkData is a reference to the version of the Secret containing the network data used to build the image.

Type

object

Property	Type	Description
name	string	
version	string	

11.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/metal3.io/v1alpha1/preprovisioningimages**
 - **GET**: list objects of kind PreprovisioningImage
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages**
 - **DELETE**: delete collection of PreprovisioningImage
 - **GET**: list objects of kind PreprovisioningImage
 - **POST**: create a PreprovisioningImage
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages/{name}**
 - **DELETE**: delete a PreprovisioningImage
 - **GET**: read the specified PreprovisioningImage
 - **PATCH**: partially update the specified PreprovisioningImage
 - **PUT**: replace the specified PreprovisioningImage
- **/apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages/{name}/status**
 - **GET**: read status of the specified PreprovisioningImage
 - **PATCH**: partially update status of the specified PreprovisioningImage
 - **PUT**: replace status of the specified PreprovisioningImage

11.2.1. /apis/metal3.io/v1alpha1/preprovisioningimages

HTTP method

GET

Description

list objects of kind PreprovisioningImage

Table 11.1. HTTP responses

HTTP code	Reponse body
200 - OK	PreprovisioningImageList schema

HTTP code	Reponse body
401 - Unauthorized	Empty

11.2.2. /apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages

HTTP method

DELETE

Description

delete collection of PreprovisioningImage

Table 11.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind PreprovisioningImage

Table 11.3. HTTP responses

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

HTTP method

POST

Description

create a PreprovisioningImage

Table 11.4. 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: <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 11.5. Body parameters

Parameter	Type	Description
body	PreprovisioningImage schema	

Table 11.6. HTTP responses

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

11.2.3. /apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages/{name}

Table 11.7. Global path parameters

Parameter	Type	Description
name	string	name of the PreprovisioningImage

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

delete a PreprovisioningImage

Table 11.8. 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 11.9. HTTP responses

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

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

read the specified PreprovisioningImage

Table 11.10. HTTP responses

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

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

partially update the specified PreprovisioningImage

Table 11.11. 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 11.12. HTTP responses

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

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

replace the specified PreprovisioningImage

Table 11.13. 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 11.14. Body parameters

Parameter	Type	Description
body	PreprovisioningImage schema	

Table 11.15. HTTP responses

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

11.2.4. /apis/metal3.io/v1alpha1/namespaces/{namespace}/preprovisioningimages/{name}

Table 11.16. Global path parameters

Parameter	Type	Description
name	string	name of the PreprovisioningImage

HTTP method

GET

Description

read status of the specified PreprovisioningImage

Table 11.17. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified PreprovisioningImage

Table 11.18. 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 11.19. HTTP responses

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

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

replace status of the specified PreprovisioningImage

Table 11.20. 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 11.21. Body parameters

Parameter	Type	Description
body	PreprovisioningImage schema	

Table 11.22. HTTP responses

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

CHAPTER 12. PROVISIONING [METAL3.IO/V1ALPHA1]

Description

Provisioning contains configuration used by the Provisioning service (Ironic) to provision baremetal hosts. Provisioning is created by the OpenShift installer using admin or user provided information about the provisioning network and the NIC on the server that can be used to PXE boot it. This CR is a singleton, created by the installer and currently only consumed by the cluster-baremetal-operator to bring up and update containers in a metal3 cluster.

Type

object

12.1. 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	ProvisioningSpec defines the desired state of Provisioning
status	object	ProvisioningStatus defines the observed state of Provisioning

12.1.1. .spec

Description

ProvisioningSpec defines the desired state of Provisioning

Type

object

Property	Type	Description
additionalNTPServers	array (string)	AdditionalNTPServers is a list of NTP Servers to be used by the provisioning service
bootIsoSource	string	BootIsoSource provides a way to set the location where the iso image to boot the nodes will be served from. By default the boot iso image is cached locally and served from the Provisioning service (Ironic) nodes using an auxiliary httpd server. If the boot iso image is already served by an httpd server, setting this option to http allows to directly provide the image from there; in this case, the network (either internal or external) where the httpd server that hosts the boot iso is needs to be accessible by the metal3 pod.
disableVirtualMediaTLS	boolean	DisableVirtualMediaTLS turns off TLS on the virtual media server, which may be required for hardware that cannot accept HTTPS links.
preProvisioningOSDownloadURLs	object	PreprovisioningOSDownloadURLs is set of CoreOS Live URLs that would be necessary to provision a worker either using virtual media or PXE.
provisioningDHCPExternal	boolean	ProvisioningDHCPExternal indicates whether the DHCP server for IP addresses in the provisioning DHCP range is present within the metal3 cluster or external to it. This field is being deprecated in favor of provisioningNetwork.

Property	Type	Description
provisioningDHCPRange	string	<p>ProvisioningDHCPRange needs to be interpreted along with ProvisioningDHCPExternal. If the value of provisioningDHCPExternal is set to False, then ProvisioningDHCPRange represents the range of IP addresses that the DHCP server running within the metal3 cluster can use while provisioning baremetal servers. If the value of ProvisioningDHCPExternal is set to True, then the value of ProvisioningDHCPRange will be ignored. When the value of ProvisioningDHCPExternal is set to False, indicating an internal DHCP server and the value of ProvisioningDHCPRange is not set, then the DHCP range is taken to be the default range which goes from .10 to .100 of the ProvisioningNetworkCIDR. This is the only value in all of the Provisioning configuration that can be changed after the installer has created the CR. This value needs to be two comma sererated IP addresses within the ProvisioningNetworkCIDR where the 1st address represents the start of the range and the 2nd address represents the last usable address in the range.</p>
provisioningDNS	boolean	<p>ProvisioningDNS allows sending the DNS information via DHCP on the provisionig network. It is off by default since the Provisioning service itself (Ironic) does not require DNS, but it may be useful for layered products (e.g. ZTP).</p>

Property	Type	Description
provisioningIP	string	ProvisioningIP is the IP address assigned to the provisioningInterface of the baremetal server. This IP address should be within the provisioning subnet, and outside of the DHCP range.
provisioningInterface	string	ProvisioningInterface is the name of the network interface on a baremetal server to the provisioning network. It can have values like eth1 or ens3.
provisioningMacAddresses	array (string)	ProvisioningMacAddresses is a list of mac addresses of network interfaces on a baremetal server to the provisioning network. Use this instead of ProvisioningInterface to allow interfaces of different names. If not provided it will be populated by the BMH.Spec.BootMacAddress of each master.

Property	Type	Description
provisioningNetwork	string	<p>ProvisioningNetwork provides a way to indicate the state of the underlying network configuration for the provisioning network. This field can have one of the following values - Managed- when the provisioning network is completely managed by the Baremetal IPI solution.</p> <p>Unmanaged- when the provisioning network is present and used but the user is responsible for managing DHCP. Virtual media provisioning is recommended but PXE is still available if required. Disabled- when the provisioning network is fully disabled. User can bring up the baremetal cluster using virtual media or assisted installation. If using metal3 for power management, BMCs must be accessible from the machine networks. User should provide two IPs on the external network that would be used for provisioning services.</p>
provisioningNetworkCIDR	string	<p>ProvisioningNetworkCIDR is the network on which the baremetal nodes are provisioned. The provisioningIP and the IPs in the dhcpRange all come from within this network. When using IPv6 and in a network managed by the Baremetal IPI solution this cannot be a network larger than a /64.</p>
provisioningOSDownloadURL	string	<p>ProvisioningOSDownloadURL is the location from which the OS Image used to boot baremetal host machines can be downloaded by the metal3 cluster.</p>

Property	Type	Description
virtualMediaViaExternalNetwork	boolean	VirtualMediaViaExternalNetwork flag when set to "true" allows for workers to boot via Virtual Media and contact metal3 over the External Network. When the flag is set to "false" (which is the default), virtual media deployments can still happen based on the configuration specified in the ProvisioningNetwork i.e when in Disabled mode, over the External Network and over Provisioning Network when in Managed mode. PXE deployments will always use the Provisioning Network and will not be affected by this flag.
watchAllNamespaces	boolean	WatchAllNamespaces provides a way to explicitly allow use of this Provisioning configuration across all Namespaces. It is an optional configuration which defaults to false and in that state will be used to provision baremetal hosts in only the openshift-machine-api namespace. When set to true, this provisioning configuration would be used for baremetal hosts across all namespaces.

12.1.2. .spec.preProvisioningOSDownloadURLs

Description

PreprovisioningOSDownloadURLs is set of CoreOS Live URLs that would be necessary to provision a worker either using virtual media or PXE.

Type

object

Property	Type	Description
initramfsURL	string	InitramfsURL Image URL to be used for PXE deployments
isoURL	string	IsoURL Image URL to be used for Live ISO deployments

Property	Type	Description
kernelURL	string	KernelURL is an Image URL to be used for PXE deployments
rootfsURL	string	RootfsURL Image URL to be used for PXE deployments

12.1.3. .status

Description

ProvisioningStatus defines the observed state of Provisioning

Type

object

Property	Type	Description
conditions	array	conditions is a list of conditions and their status
conditions[]	object	OperatorCondition is just the standard condition fields.
generations	array	generations are used to determine when an item needs to be reconciled or has changed in a way that needs a reaction.
generations[]	object	GenerationStatus keeps track of the generation for a given resource so that decisions about forced updates can be made.
observedGeneration	integer	observedGeneration is the last generation change you've dealt with
readyReplicas	integer	readyReplicas indicates how many replicas are ready and at the desired state
version	string	version is the level this availability applies to

12.1.4. .status.conditions

Description

conditions is a list of conditions and their status

Type

array

12.1.5. .status.conditions[]

Description

OperatorCondition is just the standard condition fields.

Type

object

Required

- **type**

Property	Type	Description
lastTransitionTime	string	
message	string	
reason	string	
status	string	
type	string	

12.1.6. .status.generations

Description

generations are used to determine when an item needs to be reconciled or has changed in a way that needs a reaction.

Type

array

12.1.7. .status.generations[]

Description

GenerationStatus keeps track of the generation for a given resource so that decisions about forced updates can be made.

Type

object

Property	Type	Description
group	string	group is the group of the thing you're tracking

Property	Type	Description
hash	string	hash is an optional field set for resources without generation that are content sensitive like secrets and configmaps
lastGeneration	integer	lastGeneration is the last generation of the workload controller involved
name	string	name is the name of the thing you're tracking
namespace	string	namespace is where the thing you're tracking is
resource	string	resource is the resource type of the thing you're tracking

12.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/metal3.io/v1alpha1/provisionings**
 - **DELETE**: delete collection of Provisioning
 - **GET**: list objects of kind Provisioning
 - **POST**: create a Provisioning
- **/apis/metal3.io/v1alpha1/provisionings/{name}**
 - **DELETE**: delete a Provisioning
 - **GET**: read the specified Provisioning
 - **PATCH**: partially update the specified Provisioning
 - **PUT**: replace the specified Provisioning
- **/apis/metal3.io/v1alpha1/provisionings/{name}/status**
 - **GET**: read status of the specified Provisioning
 - **PATCH**: partially update status of the specified Provisioning
 - **PUT**: replace status of the specified Provisioning

12.2.1. /apis/metal3.io/v1alpha1/provisionings

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

delete collection of Provisioning

Table 12.1. HTTP responses

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

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

list objects of kind Provisioning

Table 12.2. HTTP responses

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

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

create a Provisioning

Table 12.3. 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 12.4. Body parameters

Parameter	Type	Description
body	Provisioning schema	

Table 12.5. HTTP responses

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

12.2.2. /apis/metal3.io/v1alpha1/provisionings/{name}

Table 12.6. Global path parameters

Parameter	Type	Description
name	string	name of the Provisioning

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

delete a Provisioning

Table 12.7. 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 12.8. HTTP responses

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

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

read the specified Provisioning

Table 12.9. HTTP responses

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

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

partially update the specified Provisioning

Table 12.10. 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 12.11. HTTP responses

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

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

replace the specified Provisioning

Table 12.12. 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 12.13. Body parameters

Parameter	Type	Description
body	Provisioning schema	

Table 12.14. HTTP responses

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

12.2.3. /apis/metal3.io/v1alpha1/provisionings/{name}/status

Table 12.15. Global path parameters

Parameter	Type	Description
name	string	name of the Provisioning

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

read status of the specified Provisioning

Table 12.16. HTTP responses

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

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

partially update status of the specified Provisioning

Table 12.17. 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 12.18. HTTP responses

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

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

replace status of the specified Provisioning

Table 12.19. 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 12.20. Body parameters

Parameter	Type	Description
body	Provisioning schema	

Table 12.21. HTTP responses

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