OpenShift Container Platform 4.18

Config APIs

Reference guide for config APIs

Last Updated: 2025-07-25

OpenShift Container Platform 4.18 Config APIs

Reference guide for config APIs

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. CONFIG APIS

1.1. APISERVER [CONFIG.OPENSHIFT.IO/V1]

Description

APIServer holds configuration (like serving certificates, client CA and CORS domains) shared by all API servers in the system, among them especially kube-apiserver and openshift-apiserver. The canonical name of an instance is 'cluster'.

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

Type

object

1.2. AUTHENTICATION [CONFIG.OPENSHIFT.IO/V1]

Description

Authentication specifies cluster-wide settings for authentication (like OAuth and webhook token authenticators). The canonical name of an instance is **cluster**.

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

Type

object

1.3. BUILD [CONFIG.OPENSHIFT.IO/V1]

Description

Build configures the behavior of OpenShift builds for the entire cluster. This includes default settings that can be overridden in BuildConfig objects, and overrides which are applied to all builds.

The canonical name is "cluster"

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

Type

object

1.4. CLUSTEROPERATOR [CONFIG.OPENSHIFT.IO/V1]

Description

ClusterOperator is the Custom Resource object which holds the current state of an operator. This object is used by operators to convey their state to the rest of the cluster.

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

Type

object

1.5. CLUSTERVERSION [CONFIG.OPENSHIFT.IO/V1]

Description

ClusterVersion is the configuration for the ClusterVersionOperator. This is where parameters related to automatic updates can be set.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.6. CONSOLE [CONFIG.OPENSHIFT.IO/V1]

Description

Console holds cluster-wide configuration for the web console, including the logout URL, and reports the public URL of the console. The canonical name is **cluster**.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.7. DNS [CONFIG.OPENSHIFT.IO/V1]

Description

DNS holds cluster-wide information about DNS. The canonical name is **cluster**
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.8. FEATUREGATE [CONFIG.OPENSHIFT.IO/V1]

Description

Feature holds cluster-wide information about feature gates. The canonical name is **cluster**
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.9. HELMCHARTREPOSITORY [HELM.OPENSHIFT.IO/V1BETA1]

Description

HelmChartRepository holds cluster-wide configuration for proxied Helm chart repository
Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

1.10. IMAGE [CONFIG.OPENSHIFT.IO/V1]

Description

Image governs policies related to imagestream imports and runtime configuration for external registries. It allows cluster admins to configure which registries OpenShift is allowed to import images from, extra CA trust bundles for external registries, and policies to block or allow registry hostnames. When exposing OpenShift's image registry to the public, this also lets cluster admins specify the external hostname.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.11. IMAGEDIGESTMIRRORSET [CONFIG.OPENSHIFT.IO/V1]

Description

ImageDigestMirrorSet holds cluster-wide information about how to handle registry mirror rules on using digest pull specification. When multiple policies are defined, the outcome of the behavior is defined on each field.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.12. IMAGECONTENTPOLICY [CONFIG.OPENSHIFT.IO/V1]

Description

ImageContentPolicy holds cluster-wide information about how to handle registry mirror rules. When multiple policies are defined, the outcome of the behavior is defined on each field.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.13. IMAGETAGMIRRORSET [CONFIG.OPENSHIFT.IO/V1]

Description

ImageTagMirrorSet holds cluster-wide information about how to handle registry mirror rules on using tag pull specification. When multiple policies are defined, the outcome of the behavior is defined on each field.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.14. INFRASTRUCTURE [CONFIG.OPENSHIFT.IO/V1]

Description

Infrastructure holds cluster-wide information about Infrastructure. The canonical name is **cluster**

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

Type

object

1.15. INGRESS [CONFIG.OPENSHIFT.IO/V1]

Description

Ingress holds cluster-wide information about ingress, including the default ingress domain used for routes. The canonical name is **cluster**.

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

Type

object

1.16. NETWORK [CONFIG.OPENSHIFT.IO/V1]

Description

Network holds cluster-wide information about Network. The canonical name is **cluster**. It is used to configure the desired network configuration, such as: IP address pools for services/pod IPs, network plugin, etc. Please view network.spec for an explanation on what applies when configuring this resource.

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

Type

object

1.17. NODE [CONFIG.OPENSHIFT.IO/V1]

Description

Node holds cluster-wide information about node specific features.

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

Type

object

1.18. OAUTH [CONFIG.OPENSHIFT.IO/V1]

Description

OAuth holds cluster-wide information about OAuth. The canonical name is **cluster**. It is used to configure the integrated OAuth server. This configuration is only honored when the top level Authentication config has type set to IntegratedOAuth.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.19. OPERATORHUB [CONFIG.OPENSHIFT.IO/V1]

Description

OperatorHub is the Schema for the operatorhubs API. It can be used to change the state of the default hub sources for OperatorHub on the cluster from enabled to disabled and vice versa.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.20. PROJECT [CONFIG.OPENSHIFT.IO/V1]

Description

Project holds cluster-wide information about Project. The canonical name is **cluster**
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.21. PROJECTHELMCHARTREPOSITORY [HELM.OPENSHIFT.IO/V1BETA1]

Description

ProjectHelmChartRepository holds namespace-wide configuration for proxied Helm chart repository
Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

1.22. PROXY [CONFIG.OPENSHIFT.IO/V1]

Description

Proxy holds cluster-wide information on how to configure default proxies for the cluster. The canonical name is **cluster**
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.23. SCHEDULER [CONFIG.OPENSHIFT.IO/V1]

Description

Scheduler holds cluster-wide config information to run the Kubernetes Scheduler and influence its placement decisions. The canonical name for this config is **cluster**.

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

Type

object

CHAPTER 2. APISERVER [CONFIG.OPENSHIFT.IO/V1]

Description

APIServer holds configuration (like serving certificates, client CA and CORS domains) shared by all API servers in the system, among them especially kube-apiserver and openshift-apiserver. The canonical name of an instance is 'cluster'.

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

Type

object

Required

- **spec**

2.1. SPECIFICATION

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

Property	Type	Description
spec	**object**	spec holds user settable values for configuration
status	**object**	status holds observed values from the cluster. They may not be overridden.

2.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
additionalCORSAllowedOrigins	**array (string)**	additionalCORSAllowedOrigins lists additional, user-defined regular expressions describing hosts for which the API server allows access using the CORS headers. This may be needed to access the API and the integrated OAuth server from JavaScript applications. The values are regular expressions that correspond to the Golang regular expression language.
audit	**object**	audit specifies the settings for audit configuration to be applied to all OpenShift-provided API servers in the cluster.

Property	Type	Description
clientCA	**object**	clientCA references a ConfigMap containing a certificate bundle for the signers that will be recognized for incoming client certificates in addition to the operator managed signers. If this is empty, then only operator managed signers are valid. You usually only have to set this if you have your own PKI you wish to honor client certificates from. The ConfigMap must exist in the openshift-config namespace and contain the following required fields: - ConfigMap.Data["ca-bundle.crt"] - CA bundle.
encryption	**object**	encryption allows the configuration of encryption of resources at the datastore layer.
servingCerts	**object**	servingCert is the TLS cert info for serving secure traffic. If not specified, operator managed certificates will be used for serving secure traffic.
tlsSecurityProfile	**object**	tlsSecurityProfile specifies settings for TLS connections for externally exposed servers. If unset, a default (which may change between releases) is chosen. Note that only Old, Intermediate and Custom profiles are currently supported, and the maximum available minTLSVersion is VersionTLS12.

2.1.2. .spec.audit

Description

audit specifies the settings for audit configuration to be applied to all OpenShift-provided API servers in the cluster.

Type

object

Property	Type	Description
customRules	**array**	customRules specify profiles per group. These profile take precedence over the top-level profile field if they apply. They are evaluation from top to bottom and the first one that matches, applies.
customRules[]	**object**	AuditCustomRule describes a custom rule for an audit profile that takes precedence over the top-level profile.

Property	Type	Description
profile	**string**	profile specifies the name of the desired top-level audit profile to be applied to all requests sent to any of the OpenShift-provided API servers in the cluster (kube-apiserver, openshift-apiserver and oauth-apiserver), with the exception of those requests that match one or more of the customRules. The following profiles are provided: - Default: default policy which means MetaData level logging with the exception of events (not logged at all), oauthaccesstokens and oauthauthorizetokens (both logged at RequestBody level). - WriteRequestBodies: like 'Default', but logs request and response HTTP payloads for write requests (create, update, patch). - AllRequestBodies: like 'WriteRequestBodies', but also logs request and response HTTP payloads for read requests (get, list). - None: no requests are logged at all, not even oauthaccesstokens and oauthauthorizetokens. Warning: It is not recommended to disable audit logging by using the **None** profile unless you are fully aware of the risks of not logging data that can be beneficial when troubleshooting issues. If you disable audit logging and a support situation arises, you might need to enable audit logging and reproduce the issue in order to troubleshoot properly. If unset, the 'Default' profile is used as the default.

2.1.3. .spec.audit.customRules

Description

customRules specify profiles per group. These profile take precedence over the top-level profile field if they apply. They are evaluation from top to bottom and the first one that matches, applies.

Type

array

2.1.4. .spec.audit.customRules[]

Description

AuditCustomRule describes a custom rule for an audit profile that takes precedence over the top-level profile.

Type

object

Required

- **group**
- **profile**

Property	Type	Description
group	**string**	group is a name of group a request user must be member of in order to this profile to apply.
profile	**string**	profile specifies the name of the desired audit policy configuration to be deployed to all OpenShift-provided API servers in the cluster. The following profiles are provided: - Default: the existing default policy. - WriteRequestBodies: like 'Default', but logs request and response HTTP payloads for write requests (create, update, patch). - AllRequestBodies: like 'WriteRequestBodies', but also logs request and response HTTP payloads for read requests (get, list). - None: no requests are logged at all, not even oauthaccesstokens and oauthauthorizetokens. If unset, the 'Default' profile is used as the default.

2.1.5. .spec.clientCA

Description

clientCA references a ConfigMap containing a certificate bundle for the signers that will be recognized for incoming client certificates in addition to the operator managed signers. If this is empty, then only operator managed signers are valid. You usually only have to set this if you have your own PKI you wish to honor client certificates from. The ConfigMap must exist in the openshift-config namespace and contain the following required fields: - ConfigMap.Data["ca-bundle.crt"] - CA bundle.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

2.1.6. .spec.encryption

Description

encryption allows the configuration of encryption of resources at the datastore layer.

Type

object

Property	Type	Description

Property	Type	Description
type	**string**	type defines what encryption type should be used to encrypt resources at the datastore layer. When this field is unset (i.e. when it is set to the empty string), identity is implied. The behavior of unset can and will change over time. Even if encryption is enabled by default, the meaning of unset may change to a different encryption type based on changes in best practices. When encryption is enabled, all sensitive resources shipped with the platform are encrypted. This list of sensitive resources can and will change over time. The current authoritative list is: 1. secrets 2. configmaps 3. routes.route.openshift.io 4. oauthaccesstokens.oauth.openshift.io 5. oauthauthorizetokens.oauth.openshift.io

2.1.7. .spec.servingCerts

Description

servingCert is the TLS cert info for serving secure traffic. If not specified, operator managed certificates will be used for serving secure traffic.

Type

object

Property	Type	Description
namedCertificates	**array**	namedCertificates references secrets containing the TLS cert info for serving secure traffic to specific hostnames. If no named certificates are provided, or no named certificates match the server name as understood by a client, the defaultServingCertificate will be used.

Property	Type	Description
namedCertificates[]	**object**	APIServerNamedServingCert maps a server DNS name, as understood by a client, to a certificate.

2.1.8. .spec.servingCerts.namedCertificates

Description

namedCertificates references secrets containing the TLS cert info for serving secure traffic to specific hostnames. If no named certificates are provided, or no named certificates match the server name as understood by a client, the defaultServingCertificate will be used.

Type

array

2.1.9. .spec.servingCerts.namedCertificates[]

Description

APIServerNamedServingCert maps a server DNS name, as understood by a client, to a certificate.

Type

object

Property	Type	Description
names	**array (string)**	names is a optional list of explicit DNS names (leading wildcards allowed) that should use this certificate to serve secure traffic. If no names are provided, the implicit names will be extracted from the certificates. Exact names trump over wildcard names. Explicit names defined here trump over extracted implicit names.
servingCertificate	**object**	servingCertificate references a kubernetes.io/tls type secret containing the TLS cert info for serving secure traffic. The secret must exist in the openshift-config namespace and contain the following required fields: - Secret.Data["tls.key"] - TLS private key. - Secret.Data["tls.crt"] - TLS certificate.

2.1.10. .spec.servingCerts.namedCertificates[].servingCertificate

Description

servingCertificate references a kubernetes.io/tls type secret containing the TLS cert info for serving secure traffic. The secret must exist in the openshift-config namespace and contain the following required fields: - Secret.Data["tls.key"] - TLS private key. - Secret.Data["tls.crt"] - TLS certificate.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

2.1.11. .spec.tlsSecurityProfile

Description

tlsSecurityProfile specifies settings for TLS connections for externally exposed servers.
If unset, a default (which may change between releases) is chosen. Note that only Old, Intermediate and Custom profiles are currently supported, and the maximum available minTLSVersion is VersionTLS12.

Type

object

Property	Type	Description

Property	Type	Description
custom	``	custom is a user-defined TLS security profile. Be extremely careful using a custom profile as invalid configurations can be catastrophic. An example custom profile looks like this: ciphers: - ECDHE-ECDSA-CHACHA20-POLY1305 - ECDHE-RSA-CHACHA20-POLY1305 - ECDHE-RSA-AES128-GCM-SHA256 - ECDHE-ECDSA-AES128-GCM-SHA256 minTLSVersion: VersionTLS11

Property	Type	Description
intermediate	``	intermediate is a TLS security profile based on: https://wiki.mozilla.org/Security/Server_Side_TLS#Intermediate_compatibility_.28recommended.29 and looks like this (yaml): ciphers: - TLS_AES_128_GCM_SHA256 - TLS_AES_256_GCM_SHA384 - TLS_CHACHA20_POLY1305_SHA256 - ECDHE-ECDSA-AES128-GCM-SHA256 - ECDHE-RSA-AES128-GCM-SHA256 - ECDHE-ECDSA-AES256-GCM-SHA384 - ECDHE-RSA-AES256-GCM-SHA384 - ECDHE-ECDSA-CHACHA20-POLY1305 - ECDHE-RSA-CHACHA20-POLY1305 - DHE-RSA-AES128-GCM-SHA256 - DHE-RSA-AES256-GCM-SHA384 minTLSVersion: VersionTLS12

Property	Type	Description
modern	``	modern is a TLS security profile based on: https://wiki.mozilla.org/Security/Server_Side_TLS#Modern_compatibility and looks like this (yaml): ciphers: - TLS_AES_128_GCM_SHA256 - TLS_AES_256_GCM_SHA384 - TLS_CHACHA20_POLY1305_SHA256 minTLSVersion: VersionTLS13
old	``	old is a TLS security profile based on: https://wiki.mozilla.org/Security/Server_Side_TLS#Old_backward_compatibility and looks like this (yaml): ciphers: - TLS_AES_128_GCM_SHA256 - TLS_AES_256_GCM_SHA384 - TLS_CHACHA20_POLY1305_SHA256 - ECDHE-ECDSA-AES128-GCM-SHA256 - ECDHE-RSA-AES128-GCM-SHA256 - ECDHE-ECDSA-AES256-GCM-SHA384 - ECDHE-RSA-AES256-GCM-SHA384 - ECDHE-ECDSA-CHACHA20-POLY1305

Property	Type	Description
		- ECDHE-RSA-CHACHA20-POLY1305 - DHE-RSA-AES128-GCM-SHA256 - DHE-RSA-AES256-GCM-SHA384 - DHE-RSA-CHACHA20-POLY1305 - ECDHE-ECDSA-AES128-SHA256 - ECDHE-RSA-AES128-SHA256 - ECDHE-ECDSA-AES128-SHA - ECDHE-RSA-AES128-SHA - ECDHE-ECDSA-AES256-SHA384 - ECDHE-RSA-AES256-SHA384 - ECDHE-ECDSA-AES256-SHA - ECDHE-RSA-AES256-SHA - DHE-RSA-AES128-SHA256 - DHE-RSA-AES256-SHA256 - AES128-GCM-SHA256 - AES256-GCM-SHA384 - AES128-SHA256 - AES256-SHA256 - AES128-SHA - AES256-SHA - DES-CBC3-SHA

minTLSVersion: VersionTLS10

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

2.1.12. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/apiservers**
  - **DELETE**: delete collection of APIServer
  - **GET**: list objects of kind APIServer
  - **POST**: create an APIServer
- **/apis/config.openshift.io/v1/apiservers/{name}**
  - **DELETE**: delete an APIServer
  - **GET**: read the specified APIServer
  - **PATCH**: partially update the specified APIServer

- **PUT**: replace the specified APIServer

- **/apis/config.openshift.io/v1/apiservers/{name}/status**
  - **GET**: read status of the specified APIServer
  - **PATCH**: partially update status of the specified APIServer
  - **PUT**: replace status of the specified APIServer

2.2.1. /apis/config.openshift.io/v1/apiservers

HTTP method

DELETE

Description

delete collection of APIServer

Table 2.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind APIServer

Table 2.2. HTTP responses

HTTP code	Reponse body
200 - OK	**APIServerList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an APIServer

Table 2.3. Query parameters

Parameter	Type	Description

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

Table 2.4. Body parameters

Parameter	Type	Description
body	**APIServer** schema	

Table 2.5. HTTP responses

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

2.2.2. /apis/config.openshift.io/v1/apiservers/{name}

Table 2.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the APIServer

HTTP method

DELETE

Description

delete an APIServer

Table 2.7. Query parameters

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

Table 2.8. HTTP responses

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

HTTP method

GET

Description

read the specified APIServer

Table 2.9. HTTP responses

HTTP code	Reponse body
200 - OK	**APIServer** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified APIServer

Table 2.10. Query parameters

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

Table 2.11. HTTP responses

HTTP code	Reponse body
200 - OK	**APIServer** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified APIServer

Table 2.12. Query parameters

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

Table 2.13. Body parameters

Parameter	Type	Description
body	**APIServer** schema	

Table 2.14. HTTP responses

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

2.2.3. /apis/config.openshift.io/v1/apiservers/{name}/status

Table 2.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the APIServer

HTTP method

GET

Description

read status of the specified APIServer

Table 2.16. HTTP responses

HTTP code	Reponse body
200 - OK	**APIServer** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified APIServer

Table 2.17. Query parameters

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

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

Table 2.18. HTTP responses

HTTP code	Reponse body
200 - OK	**APIServer** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified APIServer

Table 2.19. Query parameters

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

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

Table 2.20. Body parameters

Parameter	Type	Description
body	**APIServer** schema	

Table 2.21. HTTP responses

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

CHAPTER 3. AUTHENTICATION [CONFIG.OPENSHIFT.IO/V1]

Description

Authentication specifies cluster-wide settings for authentication (like OAuth and webhook token authenticators). The canonical name of an instance is **cluster**.

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

Type

object

Required

- **spec**

3.1. SPECIFICATION

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

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

3.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
oauthMetadata	**object**	oauthMetadata contains the discovery endpoint data for OAuth 2.0 Authorization Server Metadata for an external OAuth server. This discovery document can be viewed from its served location: oc get --raw '/.well-known/oauth-authorization-server' For further details, see the IETF Draft: https://tools.ietf.org/html/draft-ietf-oauth-discovery-04#section-2 If oauthMetadata.name is non-empty, this value has precedence over any metadata reference stored in status. The key "oauthMetadata" is used to locate the data. If specified and the config map or expected key is not found, no metadata is served. If the specified metadata is not valid, no metadata is served. The namespace for this config map is openshift-config.

Property	Type	Description
serviceAccountIssuer	**string**	serviceAccountIssuer is the identifier of the bound service account token issuer. The default is https://kubernetes.default.svc WARNING: Updating this field will not result in immediate invalidation of all bound tokens with the previous issuer value. Instead, the tokens issued by previous service account issuer will continue to be trusted for a time period chosen by the platform (currently set to 24h). This time period is subject to change over time. This allows internal components to transition to use new service account issuer without service distruption.
type	**string**	type identifies the cluster managed, user facing authentication mode in use. Specifically, it manages the component that responds to login attempts. The default is IntegratedOAuth.
webhookTokenAuthenticator	**object**	webhookTokenAuthenticator configures a remote token reviewer. These remote authentication webhooks can be used to verify bearer tokens via the tokenreviews.authentication.k8s.io REST API. This is required to honor bearer tokens that are provisioned by an external authentication service. Can only be set if "Type" is set to "None".
webhookTokenAuthenticators	**array**	webhookTokenAuthenticators is DEPRECATED, setting it has no effect.

Property	Type	Description
webhookTokenAuthenticators[]	**object**	deprecatedWebhookTokenAuthenticator holds the necessary configuration options for a remote token authenticator. It's the same as WebhookTokenAuthenticator but it's missing the 'required' validation on KubeConfig field.

3.1.2. .spec.oauthMetadata

Description

oauthMetadata contains the discovery endpoint data for OAuth 2.0 Authorization Server Metadata for an external OAuth server. This discovery document can be viewed from its served location: oc get --raw '/.well-known/oauth-authorization-server' For further details, see the IETF Draft: https://tools.ietf.org/html/draft-ietf-oauth-discovery-04#section-2 If oauthMetadata.name is non-empty, this value has precedence over any metadata reference stored in status. The key "oauthMetadata" is used to locate the data. If specified and the config map or expected key is not found, no metadata is served. If the specified metadata is not valid, no metadata is served. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

3.1.3. .spec.webhookTokenAuthenticator

Description

webhookTokenAuthenticator configures a remote token reviewer. These remote authentication webhooks can be used to verify bearer tokens via the tokenreviews.authentication.k8s.io REST API. This is required to honor bearer tokens that are provisioned by an external authentication service. Can only be set if "Type" is set to "None".

Type

object

Required

- **kubeConfig**

Property	Type	Description
kubeConfig	**object**	kubeConfig references a secret that contains kube config file data which describes how to access the remote webhook service. The namespace for the referenced secret is openshift-config. For further details, see: https://kubernetes.io/docs/reference/access-authn-authz/authentication/#webhook-token-authentication The key "kubeConfig" is used to locate the data. If the secret or expected key is not found, the webhook is not honored. If the specified kube config data is not valid, the webhook is not honored.

3.1.4. .spec.webhookTokenAuthenticator.kubeConfig

Description

kubeConfig references a secret that contains kube config file data which describes how to access the remote webhook service. The namespace for the referenced secret is openshift-config.
For further details, see:

https://kubernetes.io/docs/reference/access-authn-authz/authentication/#webhook-token-authentication

The key "kubeConfig" is used to locate the data. If the secret or expected key is not found, the webhook is not honored. If the specified kube config data is not valid, the webhook is not honored.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

3.1.5. .spec.webhookTokenAuthenticators

Description

webhookTokenAuthenticators is DEPRECATED, setting it has no effect.

Type

array

3.1.6. .spec.webhookTokenAuthenticators[]

Description

deprecatedWebhookTokenAuthenticator holds the necessary configuration options for a remote token authenticator. It's the same as WebhookTokenAuthenticator but it's missing the 'required' validation on KubeConfig field.

Type

object

Property	Type	Description
kubeConfig	**object**	kubeConfig contains kube config file data which describes how to access the remote webhook service. For further details, see: https://kubernetes.io/docs/reference/access-authn-authz/authentication/#webhook-token-authentication The key "kubeConfig" is used to locate the data. If the secret or expected key is not found, the webhook is not honored. If the specified kube config data is not valid, the webhook is not honored. The namespace for this secret is determined by the point of use.

3.1.7. .spec.webhookTokenAuthenticators[].kubeConfig

Description

kubeConfig contains kube config file data which describes how to access the remote webhook service. For further details, see: https://kubernetes.io/docs/reference/access-authn-authz/authentication/#webhook-token-authentication The key "kubeConfig" is used to locate the data. If the secret or expected key is not found, the webhook is not honored. If the specified kube config data is not valid, the webhook is not honored. The namespace for this secret is determined by the point of use.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

3.1.8. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

Property	Type	Description
integratedOAuthMetadata	**object**	integratedOAuthMetadata contains the discovery endpoint data for OAuth 2.0 Authorization Server Metadata for the in-cluster integrated OAuth server. This discovery document can be viewed from its served location: oc get --raw '/.well-known/oauth-authorization-server' For further details, see the IETF Draft: https://tools.ietf.org/html/draft-ietf-oauth-discovery-04#section-2 This contains the observed value based on cluster state. An explicitly set value in spec.oauthMetadata has precedence over this field. This field has no meaning if authentication spec.type is not set to IntegratedOAuth. The key "oauthMetadata" is used to locate the data. If the config map or expected key is not found, no metadata is served. If the specified metadata is not valid, no metadata is served. The namespace for this config map is openshift-config-managed.

3.1.9. .status.integratedOAuthMetadata

Description

integratedOAuthMetadata contains the discovery endpoint data for OAuth 2.0 Authorization Server Metadata for the in-cluster integrated OAuth server. This discovery document can be viewed from its served location: oc get --raw '/.well-known/oauth-authorization-server' For further details, see

the IETF Draft: https://tools.ietf.org/html/draft-ietf-oauth-discovery-04#section-2 This contains the observed value based on cluster state. An explicitly set value in spec.oauthMetadata has precedence over this field. This field has no meaning if authentication spec.type is not set to IntegratedOAuth. The key "oauthMetadata" is used to locate the data. If the config map or expected key is not found, no metadata is served. If the specified metadata is not valid, no metadata is served. The namespace for this config map is openshift-config-managed.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/authentications**
  - **DELETE**: delete collection of Authentication
  - **GET**: list objects of kind Authentication
  - **POST**: create an Authentication
- **/apis/config.openshift.io/v1/authentications/{name}**
  - **DELETE**: delete an Authentication
  - **GET**: read the specified Authentication
  - **PATCH**: partially update the specified Authentication
  - **PUT**: replace the specified Authentication
- **/apis/config.openshift.io/v1/authentications/{name}/status**
  - **GET**: read status of the specified Authentication
  - **PATCH**: partially update status of the specified Authentication
  - **PUT**: replace status of the specified Authentication

3.2.1. /apis/config.openshift.io/v1/authentications

HTTP method

DELETE

Description

delete collection of Authentication

Table 3.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Authentication

Table 3.2. HTTP responses

HTTP code	Reponse body
200 - OK	**AuthenticationList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an Authentication

Table 3.3. Query parameters

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

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

Table 3.4. Body parameters

Parameter	Type	Description
body	**Authentication** schema	

Table 3.5. HTTP responses

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

3.2.2. /apis/config.openshift.io/v1/authentications/{name}

Table 3.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Authentication

HTTP method

DELETE

Description

delete an Authentication

Table 3.7. Query parameters

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

Table 3.8. HTTP responses

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

HTTP method

GET

Description

read the specified Authentication

Table 3.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Authentication** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Authentication

Table 3.10. Query parameters

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

Table 3.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Authentication** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Authentication

Table 3.12. Query parameters

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

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

Table 3.13. Body parameters

Parameter	Type	Description
body	**Authentication** schema	

Table 3.14. HTTP responses

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

3.2.3. /apis/config.openshift.io/v1/authentications/{name}/status

Table 3.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Authentication

HTTP method

GET

Description

read status of the specified Authentication

Table 3.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Authentication** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Authentication

Table 3.17. Query parameters

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

HTTP code	Reponse body
200 - OK	**Authentication** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Authentication

Table 3.19. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - 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 3.20. Body parameters

Parameter	Type	Description
body	**Authentication** schema	

Table 3.21. HTTP responses

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

CHAPTER 4. BUILD [CONFIG.OPENSHIFT.IO/V1]

Description

Build configures the behavior of OpenShift builds for the entire cluster. This includes default settings that can be overridden in BuildConfig objects, and overrides which are applied to all builds.
The canonical name is "cluster"

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

Type

object

Required

- **spec**

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

Property	Type	Description
spec	**object**	Spec holds user-settable values for the build controller configuration

4.1.1. .spec

Description

Spec holds user-settable values for the build controller configuration

Type

object

Property	Type	Description
additionalTrustedCA	**object**	AdditionalTrustedCA is a reference to a ConfigMap containing additional CAs that should be trusted for image pushes and pulls during builds. The namespace for this config map is openshift-config. DEPRECATED: Additional CAs for image pull and push should be set on image.config.openshift.io/cluster instead.
buildDefaults	**object**	BuildDefaults controls the default information for Builds
buildOverrides	**object**	BuildOverrides controls override settings for builds

4.1.2. .spec.additionalTrustedCA

Description

AdditionalTrustedCA is a reference to a ConfigMap containing additional CAs that should be trusted for image pushes and pulls during builds. The namespace for this config map is openshift-config. DEPRECATED: Additional CAs for image pull and push should be set on image.config.openshift.io/cluster instead.

Type

object

Required

- **name**

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

4.1.3. .spec.buildDefaults

Description

BuildDefaults controls the default information for Builds

Type

object

Property	Type	Description
defaultProxy	**object**	DefaultProxy contains the default proxy settings for all build operations, including image pull/push and source download. Values can be overrode by setting the **HTTP_PROXY**, **HTTPS_PROXY**, and **NO_PROXY** environment variables in the build config's strategy.
env	**array**	Env is a set of default environment variables that will be applied to the build if the specified variables do not exist on the build
env[]	**object**	EnvVar represents an environment variable present in a Container.
gitProxy	**object**	GitProxy contains the proxy settings for git operations only. If set, this will override any Proxy settings for all git commands, such as git clone. Values that are not set here will be inherited from DefaultProxy.

Property	Type	Description
imageLabels	**array**	ImageLabels is a list of docker labels that are applied to the resulting image. User can override a default label by providing a label with the same name in their Build/BuildConfig.
imageLabels[]	**object**	
resources	**object**	Resources defines resource requirements to execute the build.

4.1.4. .spec.buildDefaults.defaultProxy

Description

DefaultProxy contains the default proxy settings for all build operations, including image pull/push and source download.
Values can be overrode by setting the **HTTP_PROXY**, **HTTPS_PROXY**, and **NO_PROXY** environment variables in the build config's strategy.

Type

object

Property	Type	Description
httpProxy	**string**	httpProxy is the URL of the proxy for HTTP requests. Empty means unset and will not result in an env var.
httpsProxy	**string**	httpsProxy is the URL of the proxy for HTTPS requests. Empty means unset and will not result in an env var.
noProxy	**string**	noProxy is a comma-separated list of hostnames and/or CIDRs and/or IPs for which the proxy should not be used. Empty means unset and will not result in an env var.
readinessEndpoints	**array (string)**	readinessEndpoints is a list of endpoints used to verify readiness of the proxy.

Property	Type	Description
trustedCA	**object**	trustedCA is a reference to a ConfigMap containing a CA certificate bundle. The trustedCA field should only be consumed by a proxy validator. The validator is responsible for reading the certificate bundle from the required key "ca-bundle.crt", merging it with the system default trust bundle, and writing the merged trust bundle to a ConfigMap named "trusted-ca-bundle" in the "openshift-config-managed" namespace. Clients that expect to make proxy connections must use the trusted-ca-bundle for all HTTPS requests to the proxy, and may use the trusted-ca-bundle for non-proxy HTTPS requests as well. The namespace for the ConfigMap referenced by trustedCA is "openshift-config". Here is an example ConfigMap (in yaml): apiVersion: v1 kind: ConfigMap metadata: name: user-ca-bundle namespace: openshift-config data: ca-bundle.crt: | -----BEGIN CERTIFICATE----- Custom CA certificate bundle. -----END CERTIFICATE-----

4.1.5. .spec.buildDefaults.defaultProxy.trustedCA

Description

trustedCA is a reference to a ConfigMap containing a CA certificate bundle. The trustedCA field should only be consumed by a proxy validator. The validator is responsible for reading the certificate bundle from the required key "ca-bundle.crt", merging it with the system default trust bundle, and writing the merged trust bundle to a ConfigMap named "trusted-ca-bundle" in the "openshift-config-managed" namespace. Clients that expect to make proxy connections must use the trusted-ca-bundle for all HTTPS requests to the proxy, and may use the trusted-ca-bundle for non-proxy HTTPS requests as well.
The namespace for the ConfigMap referenced by trustedCA is "openshift-config". Here is an example ConfigMap (in yaml):

apiVersion: v1 kind: ConfigMap metadata: name: user-ca-bundle namespace: openshift-config data: ca-bundle.crt: \| -----BEGIN CERTIFICATE----- Custom CA certificate bundle. -----END CERTIFICATE-----

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

4.1.6. .spec.buildDefaults.env

Description

Env is a set of default environment variables that will be applied to the build if the specified variables do not exist on the build

Type

array

4.1.7. .spec.buildDefaults.env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	Name of the environment variable. Must be a C_IDENTIFIER.

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

4.1.8. .spec.buildDefaults.env[].valueFrom

Description

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

Type

object

Property	Type	Description
configMapKeyRef	**object**	Selects a key of a ConfigMap.
fieldRef	**object**	Selects a field of the pod: supports metadata.name, metadata.namespace, **metadata.labels['<KEY>']**, **metadata.annotations['<KEY>']**, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Property	Type	Description
resourceFieldRef	**object**	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
secretKeyRef	**object**	Selects a key of a secret in the pod's namespace

4.1.9. .spec.buildDefaults.env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

object

Required

- **key**

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

4.1.10. .spec.buildDefaults.env[].valueFrom.fieldRef

Description

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

Type

object

Required

- **fieldPath**

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

4.1.11. .spec.buildDefaults.env[].valueFrom.resourceFieldRef

Description

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

Type

object

Required

- **resource**

Property	Type	Description
containerName	**string**	Container name: required for volumes, optional for env vars
divisor	**integer-or-string**	Specifies the output format of the exposed resources, defaults to "1"
resource	**string**	Required: resource to select

4.1.12. .spec.buildDefaults.env[].valueFrom.secretKeyRef

Description

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

Type

object

Required

- **key**

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

4.1.13. .spec.buildDefaults.gitProxy

Description

GitProxy contains the proxy settings for git operations only. If set, this will override any Proxy settings for all git commands, such as git clone.
Values that are not set here will be inherited from DefaultProxy.

Type

object

Property	Type	Description
httpProxy	**string**	httpProxy is the URL of the proxy for HTTP requests. Empty means unset and will not result in an env var.
httpsProxy	**string**	httpsProxy is the URL of the proxy for HTTPS requests. Empty means unset and will not result in an env var.
noProxy	**string**	noProxy is a comma-separated list of hostnames and/or CIDRs and/or IPs for which the proxy should not be used. Empty means unset and will not result in an env var.

Property	Type	Description
readinessEndpoints	**array (string)**	readinessEndpoints is a list of endpoints used to verify readiness of the proxy.
trustedCA	**object**	trustedCA is a reference to a ConfigMap containing a CA certificate bundle. The trustedCA field should only be consumed by a proxy validator. The validator is responsible for reading the certificate bundle from the required key "ca-bundle.crt", merging it with the system default trust bundle, and writing the merged trust bundle to a ConfigMap named "trusted-ca-bundle" in the "openshift-config-managed" namespace. Clients that expect to make proxy connections must use the trusted-ca-bundle for all HTTPS requests to the proxy, and may use the trusted-ca-bundle for non-proxy HTTPS requests as well. The namespace for the ConfigMap referenced by trustedCA is "openshift-config". Here is an example ConfigMap (in yaml): apiVersion: v1 kind: ConfigMap metadata: name: user-ca-bundle namespace: openshift-config data: ca-bundle.crt: | -----BEGIN CERTIFICATE----- Custom CA certificate bundle. -----END CERTIFICATE-----

4.1.14. .spec.buildDefaults.gitProxy.trustedCA

Description

trustedCA is a reference to a ConfigMap containing a CA certificate bundle. The trustedCA field should only be consumed by a proxy validator. The validator is responsible for reading the certificate bundle from the required key "ca-bundle.crt", merging it with the system default trust bundle, and writing the merged trust bundle to a ConfigMap named "trusted-ca-bundle" in the "openshift-config-managed" namespace. Clients that expect to make proxy connections must use the trusted-ca-bundle for all HTTPS requests to the proxy, and may use the trusted-ca-bundle for non-proxy HTTPS requests as well.

The namespace for the ConfigMap referenced by trustedCA is "openshift-config". Here is an example ConfigMap (in yaml):

apiVersion: v1 kind: ConfigMap metadata: name: user-ca-bundle namespace: openshift-config data: ca-bundle.crt: \| -----BEGIN CERTIFICATE----- Custom CA certificate bundle. -----END CERTIFICATE-----

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

4.1.15. .spec.buildDefaults.imageLabels

Description

ImageLabels is a list of docker labels that are applied to the resulting image. User can override a default label by providing a label with the same name in their Build/BuildConfig.

Type

array

4.1.16. .spec.buildDefaults.imageLabels[]

Description

Type

object

Property	Type	Description
name	**string**	Name defines the name of the label. It must have non-zero length.
value	**string**	Value defines the literal value of the label.

4.1.17. .spec.buildDefaults.resources

Description

Resources defines resource requirements to execute the build.

Type

object

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

4.1.18. .spec.buildDefaults.resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

4.1.19. .spec.buildDefaults.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

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

4.1.20. .spec.buildOverrides

Description

BuildOverrides controls override settings for builds

Type

object

Property	Type	Description
forcePull	**boolean**	ForcePull overrides, if set, the equivalent value in the builds, i.e. false disables force pull for all builds, true enables force pull for all builds, independently of what each build specifies itself
imageLabels	**array**	ImageLabels is a list of docker labels that are applied to the resulting image. If user provided a label in their Build/BuildConfig with the same name as one in this list, the user's label will be overwritten.
imageLabels[]	**object**	

Property	Type	Description
nodeSelector	**object (string)**	NodeSelector is a selector which must be true for the build pod to fit on a node
tolerations	**array**	Tolerations is a list of Tolerations that will override any existing tolerations set on a build pod.
tolerations[]	**object**	The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

4.1.21. .spec.buildOverrides.imageLabels

Description

ImageLabels is a list of docker labels that are applied to the resulting image. If user provided a label in their Build/BuildConfig with the same name as one in this list, the user's label will be overwritten.

Type

array

4.1.22. .spec.buildOverrides.imageLabels[]

Description

Type

object

Property	Type	Description
name	**string**	Name defines the name of the label. It must have non-zero length.
value	**string**	Value defines the literal value of the label.

4.1.23. .spec.buildOverrides.tolerations

Description

Tolerations is a list of Tolerations that will override any existing tolerations set on a build pod.

Type

array

4.1.24. .spec.buildOverrides.tolerations[]

Description

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

Type

object

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

4.2. API ENDPOINTS

The following API endpoints are available:

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

4.2.1. /apis/config.openshift.io/v1/builds

HTTP method

DELETE

Description

delete collection of Build

Table 4.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Build

Table 4.2. HTTP responses

HTTP code	Reponse body
200 - OK	**BuildList** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Build

Table 4.3. Query parameters

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

Table 4.4. Body parameters

Parameter	Type	Description
body	**Build** schema	

Table 4.5. HTTP responses

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

4.2.2. /apis/config.openshift.io/v1/builds/{name}

Table 4.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Build

HTTP method

DELETE

Description

delete a Build

Table 4.7. 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 4.8. HTTP responses

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

HTTP method

GET

Description

read the specified Build

Table 4.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Build** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Build

Table 4.10. 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 4.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Build** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Build

Table 4.12. Query parameters

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

Table 4.13. Body parameters

Parameter	Type	Description
body	**Build** schema	

Table 4.14. HTTP responses

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

4.2.3. /apis/config.openshift.io/v1/builds/{name}/status

Table 4.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Build

HTTP method

GET

Description

read status of the specified Build

Table 4.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Build** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Build

Table 4.17. 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 4.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Build** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Build

Table 4.19. 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: - 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 4.20. Body parameters

Parameter	Type	Description
body	**Build** schema	

Table 4.21. HTTP responses

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

CHAPTER 5. CLUSTEROPERATOR [CONFIG.OPENSHIFT.IO/V1]

Description

ClusterOperator is the Custom Resource object which holds the current state of an operator. This object is used by operators to convey their state to the rest of the cluster.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

5.1. SPECIFICATION

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

Property	Type	Description
spec	**object**	spec holds configuration that could apply to any operator.
status	**object**	status holds the information about the state of an operator. It is consistent with status information across the Kubernetes ecosystem.

5.1.1. .spec

Description

spec holds configuration that could apply to any operator.

Type

object

5.1.2. .status

Description

status holds the information about the state of an operator. It is consistent with status information across the Kubernetes ecosystem.

Type

object

Property	Type	Description
conditions	**array**	conditions describes the state of the operator's managed and monitored components.
conditions[]	**object**	ClusterOperatorStatusCondition represents the state of the operator's managed and monitored components.
extension	``	extension contains any additional status information specific to the operator which owns this status object.
relatedObjects	**array**	relatedObjects is a list of objects that are "interesting" or related to this operator. Common uses are: 1. the detailed resource driving the operator 2. operator namespaces 3. operand namespaces

Property	Type	Description
relatedObjects[]	**object**	ObjectReference contains enough information to let you inspect or modify the referred object.
versions	**array**	versions is a slice of operator and operand version tuples. Operators which manage multiple operands will have multiple operand entries in the array. Available operators must report the version of the operator itself with the name "operator". An operator reports a new "operator" version when it has rolled out the new version to all of its operands.
versions[]	**object**	

5.1.3. .status.conditions

Description

conditions describes the state of the operator's managed and monitored components.

Type

array

5.1.4. .status.conditions[]

Description

ClusterOperatorStatusCondition represents the state of the operator's managed and monitored components.

Type

object

Required

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

Property	Type	Description

Property	Type	Description
lastTransitionTime	**string**	lastTransitionTime is the time of the last update to the current status property.
message	**string**	message provides additional information about the current condition. This is only to be consumed by humans. It may contain Line Feed characters (U+000A), which should be rendered as new lines.
reason	**string**	reason is the CamelCase reason for the condition's current status.
status	**string**	status of the condition, one of True, False, Unknown.
type	**string**	type specifies the aspect reported by this condition.

5.1.5. .status.relatedObjects

Description

relatedObjects is a list of objects that are "interesting" or related to this operator. Common uses are: 1. the detailed resource driving the operator 2. operator namespaces 3. operand namespaces

Type

array

5.1.6. .status.relatedObjects[]

Description

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

Type

object

Required

- **group**
- **name**
- **resource**

Property	Type	Description
group	**string**	group of the referent.
name	**string**	name of the referent.
namespace	**string**	namespace of the referent.
resource	**string**	resource of the referent.

5.1.7. .status.versions

Description

versions is a slice of operator and operand version tuples. Operators which manage multiple operands will have multiple operand entries in the array. Available operators must report the version of the operator itself with the name "operator". An operator reports a new "operator" version when it has rolled out the new version to all of its operands.

Type

array

5.1.8. .status.versions[]

Description

Type

object

Required

- **name**
- **version**

Property	Type	Description
name	**string**	name is the name of the particular operand this version is for. It usually matches container images, not operators.
version	**string**	version indicates which version of a particular operand is currently being managed. It must always match the Available operand. If 1.0.0 is Available, then this must indicate 1.0.0 even if the operator is trying to rollout 1.1.0

5.2. API ENDPOINTS

The following API endpoints are available:

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

5.2.1. /apis/config.openshift.io/v1/clusteroperators

HTTP method

DELETE

Description

delete collection of ClusterOperator

Table 5.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind ClusterOperator

Table 5.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterOperatorList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a ClusterOperator

Table 5.3. Query parameters

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

Table 5.4. Body parameters

Parameter	Type	Description
body	**ClusterOperator** schema	

Table 5.5. HTTP responses

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

5.2.2. /apis/config.openshift.io/v1/clusteroperators/{name}

Table 5.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the ClusterOperator

HTTP method

DELETE

Description

delete a ClusterOperator

Table 5.7. Query parameters

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

Table 5.8. HTTP responses

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

HTTP method

GET

Description

read the specified ClusterOperator

Table 5.9. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterOperator** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ClusterOperator

Table 5.10. Query parameters

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

Table 5.11. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterOperator** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ClusterOperator

Table 5.12. Query parameters

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

Table 5.13. Body parameters

Parameter	Type	Description
body	**ClusterOperator** schema	

Table 5.14. HTTP responses

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

5.2.3. /apis/config.openshift.io/v1/clusteroperators/{name}/status

Table 5.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the ClusterOperator

HTTP method

GET

Description

read status of the specified ClusterOperator

Table 5.16. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterOperator** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ClusterOperator

Table 5.17. Query parameters

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

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

Table 5.18. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterOperator** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ClusterOperator

Table 5.19. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - 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 5.20. Body parameters

Parameter	Type	Description
body	**ClusterOperator** schema	

Table 5.21. HTTP responses

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

CHAPTER 6. CLUSTERVERSION [CONFIG.OPENSHIFT.IO/V1]

Description

ClusterVersion is the configuration for the ClusterVersionOperator. This is where parameters related to automatic updates can be set.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

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

Property	Type	Description
spec	**object**	spec is the desired state of the cluster version - the operator will work to ensure that the desired version is applied to the cluster.
status	**object**	status contains information about the available updates and any in-progress updates.

6.1.1. .spec

Description

spec is the desired state of the cluster version - the operator will work to ensure that the desired version is applied to the cluster.

Type

object

Required

- **clusterID**

Property	Type	Description
capabilities	**object**	capabilities configures the installation of optional, core cluster components. A null value here is identical to an empty object; see the child properties for default semantics.
channel	**string**	channel is an identifier for explicitly requesting that a non-default set of updates be applied to this cluster. The default channel will be contain stable updates that are appropriate for production clusters.
clusterID	**string**	clusterID uniquely identifies this cluster. This is expected to be an RFC4122 UUID value (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx in hexadecimal values). This is a required field.

Property	Type	Description
desiredUpdate	**object**	desiredUpdate is an optional field that indicates the desired value of the cluster version. Setting this value will trigger an upgrade (if the current version does not match the desired version). The set of recommended update values is listed as part of available updates in status, and setting values outside that range may cause the upgrade to fail. Some of the fields are inter-related with restrictions and meanings described here. 1. image is specified, version is specified, architecture is specified. API validation error. 2. image is specified, version is specified, architecture is not specified. You should not do this. version is silently ignored and image is used. 3. image is specified, version is not specified, architecture is specified. API validation error. 4. image is specified, version is not specified, architecture is not specified. image is used. 5. image is not specified, version is specified, architecture is specified. version and desired architecture are used to select an image. 6. image is not specified, version is specified, architecture is not specified. version and current architecture are used to select an image. 7. image is not specified, version is not specified, architecture is specified. API validation error. 8. image is not specified, version is not specified, architecture is not specified. API validation error. If an upgrade fails the operator will halt and report status about the failing component. Setting the desired update value back to the previous version will cause a rollback to be attempted. Not all rollbacks will succeed.

Property	Type	Description
overrides	**array**	overrides is list of overides for components that are managed by cluster version operator. Marking a component unmanaged will prevent the operator from creating or updating the object.
overrides[]	**object**	ComponentOverride allows overriding cluster version operator's behavior for a component.
upstream	**string**	upstream may be used to specify the preferred update server. By default it will use the appropriate update server for the cluster and region.

6.1.2. .spec.capabilities

Description

capabilities configures the installation of optional, core cluster components. A null value here is identical to an empty object; see the child properties for default semantics.

Type

object

Property	Type	Description
additionalEnabledCapabilities	**array (string)**	additionalEnabledCapabilities extends the set of managed capabilities beyond the baseline defined in baselineCapabilitySet. The default is an empty set.
baselineCapabilitySet	**string**	baselineCapabilitySet selects an initial set of optional capabilities to enable, which can be extended via additionalEnabledCapabilities. If unset, the cluster will choose a default, and the default may change over time. The current default is vCurrent.

6.1.3. .spec.desiredUpdate

Description

desiredUpdate is an optional field that indicates the desired value of the cluster version. Setting this

value will trigger an upgrade (if the current version does not match the desired version). The set of recommended update values is listed as part of available updates in status, and setting values outside that range may cause the upgrade to fail.
Some of the fields are inter-related with restrictions and meanings described here. 1. image is specified, version is specified, architecture is specified. API validation error. 2. image is specified, version is specified, architecture is not specified. You should not do this. version is silently ignored and image is used. 3. image is specified, version is not specified, architecture is specified. API validation error. 4. image is specified, version is not specified, architecture is not specified. image is used. 5. image is not specified, version is specified, architecture is specified. version and desired architecture are used to select an image. 6. image is not specified, version is specified, architecture is not specified. version and current architecture are used to select an image. 7. image is not specified, version is not specified, architecture is specified. API validation error. 8. image is not specified, version is not specified, architecture is not specified. API validation error.

If an upgrade fails the operator will halt and report status about the failing component. Setting the desired update value back to the previous version will cause a rollback to be attempted. Not all rollbacks will succeed.

Type

object

Property	Type	Description
architecture	**string**	architecture is an optional field that indicates the desired value of the cluster architecture. In this context cluster architecture means either a single architecture or a multi architecture. architecture can only be set to Multi thereby only allowing updates from single to multi architecture. If architecture is set, image cannot be set and version must be set. Valid values are 'Multi' and empty.
force	**boolean**	force allows an administrator to update to an image that has failed verification or upgradeable checks. This option should only be used when the authenticity of the provided image has been verified out of band because the provided image will run with full administrative access to the cluster. Do not use this flag with images that comes from unknown or potentially malicious sources.

Property	Type	Description
image	**string**	image is a container image location that contains the update. image should be used when the desired version does not exist in availableUpdates or history. When image is set, version is ignored. When image is set, version should be empty. When image is set, architecture cannot be specified.
version	**string**	version is a semantic version identifying the update version. version is ignored if image is specified and required if architecture is specified.

6.1.4. .spec.overrides

Description

overrides is list of overides for components that are managed by cluster version operator. Marking a component unmanaged will prevent the operator from creating or updating the object.

Type

array

6.1.5. .spec.overrides[]

Description

ComponentOverride allows overriding cluster version operator's behavior for a component.

Type

object

Required

- **group**
- **kind**
- **name**
- **namespace**
- **unmanaged**

Property	Type	Description
group	**string**	group identifies the API group that the kind is in.

Property	Type	Description
kind	**string**	kind indentifies which object to override.
name	**string**	name is the component's name.
namespace	**string**	namespace is the component's namespace. If the resource is cluster scoped, the namespace should be empty.
unmanaged	**boolean**	unmanaged controls if cluster version operator should stop managing the resources in this cluster. Default: false

6.1.6. .status

Description

status contains information about the available updates and any in-progress updates.

Type

object

Required

- **desired**
- **observedGeneration**
- **versionHash**

Property	Type	Description
availableUpdates	``	availableUpdates contains updates recommended for this cluster. Updates which appear in conditionalUpdates but not in availableUpdates may expose this cluster to known issues. This list may be empty if no updates are recommended, if the update service is unavailable, or if an invalid channel has been specified.
capabilities	**object**	capabilities describes the state of optional, core cluster components.

Property	Type	Description
conditionalUpdates	**array**	conditionalUpdates contains the list of updates that may be recommended for this cluster if it meets specific required conditions. Consumers interested in the set of updates that are actually recommended for this cluster should use availableUpdates. This list may be empty if no updates are recommended, if the update service is unavailable, or if an empty or invalid channel has been specified.
conditionalUpdates[]	**object**	ConditionalUpdate represents an update which is recommended to some clusters on the version the current cluster is reconciling, but which may not be recommended for the current cluster.
conditions	**array**	conditions provides information about the cluster version. The condition "Available" is set to true if the desiredUpdate has been reached. The condition "Progressing" is set to true if an update is being applied. The condition "Degraded" is set to true if an update is currently blocked by a temporary or permanent error. Conditions are only valid for the current desiredUpdate when metadata.generation is equal to status.generation.
conditions[]	**object**	ClusterOperatorStatusCondition represents the state of the operator's managed and monitored components.
desired	**object**	desired is the version that the cluster is reconciling towards. If the cluster is not yet fully initialized desired will be set with the information available, which may be an image or a tag.

Property	Type	Description
history	**array**	history contains a list of the most recent versions applied to the cluster. This value may be empty during cluster startup, and then will be updated when a new update is being applied. The newest update is first in the list and it is ordered by recency. Updates in the history have state Completed if the rollout completed - if an update was failing or halfway applied the state will be Partial. Only a limited amount of update history is preserved.
history[]	**object**	UpdateHistory is a single attempted update to the cluster.
observedGeneration	**integer**	observedGeneration reports which version of the spec is being synced. If this value is not equal to metadata.generation, then the desired and conditions fields may represent a previous version.
versionHash	**string**	versionHash is a fingerprint of the content that the cluster will be updated with. It is used by the operator to avoid unnecessary work and is for internal use only.

6.1.7. .status.capabilities

Description

capabilities describes the state of optional, core cluster components.

Type

object

Property	Type	Description
enabledCapabilities	**array (string)**	enabledCapabilities lists all the capabilities that are currently managed.
knownCapabilities	**array (string)**	knownCapabilities lists all the capabilities known to the current cluster.

6.1.8. .status.conditionalUpdates

Description

conditionalUpdates contains the list of updates that may be recommended for this cluster if it meets specific required conditions. Consumers interested in the set of updates that are actually recommended for this cluster should use availableUpdates. This list may be empty if no updates are recommended, if the update service is unavailable, or if an empty or invalid channel has been specified.

Type

array

6.1.9. .status.conditionalUpdates[]

Description

ConditionalUpdate represents an update which is recommended to some clusters on the version the current cluster is reconciling, but which may not be recommended for the current cluster.

Type

object

Required

- **release**
- **risks**

Property	Type	Description
conditions	**array**	conditions represents the observations of the conditional update's current status. Known types are: * Recommended, for whether the update is recommended for the current cluster.
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.
release	**object**	release is the target of the update.
risks	**array**	risks represents the range of issues associated with updating to the target release. The cluster-version operator will evaluate all entries, and only recommend the update if there is at least one entry and all entries recommend the update.

Property	Type	Description
risks[]	**object**	ConditionalUpdateRisk represents a reason and cluster-state for not recommending a conditional update.

6.1.10. .status.conditionalUpdates[].conditions

Description

conditions represents the observations of the conditional update's current status. Known types are: * Recommended, for whether the update is recommended for the current cluster.

Type

array

6.1.11. .status.conditionalUpdates[].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.

6.1.12. .status.conditionalUpdates[].release

Description

release is the target of the update.

Type

object

Required

- **image**
- **version**

Property	Type	Description
channels	**array (string)**	channels is the set of Cincinnati channels to which the release currently belongs.

Property	Type	Description
image	**string**	image is a container image location that contains the update. When this field is part of spec, image is optional if version is specified and the availableUpdates field contains a matching version.
url	**string**	url contains information about this release. This URL is set by the 'url' metadata property on a release or the metadata returned by the update API and should be displayed as a link in user interfaces. The URL field may not be set for test or nightly releases.
version	**string**	version is a semantic version identifying the update version. When this field is part of spec, version is optional if image is specified.

6.1.13. .status.conditionalUpdates[].risks

Description

risks represents the range of issues associated with updating to the target release. The cluster-version operator will evaluate all entries, and only recommend the update if there is at least one entry and all entries recommend the update.

Type

array

6.1.14. .status.conditionalUpdates[].risks[]

Description

ConditionalUpdateRisk represents a reason and cluster-state for not recommending a conditional update.

Type

object

Required

- **matchingRules**
- **message**
- **name**

- **url**

Property	Type	Description
matchingRules	**array**	matchingRules is a slice of conditions for deciding which clusters match the risk and which do not. The slice is ordered by decreasing precedence. The cluster-version operator will walk the slice in order, and stop after the first it can successfully evaluate. If no condition can be successfully evaluated, the update will not be recommended.
matchingRules[]	**object**	ClusterCondition is a union of typed cluster conditions. The 'type' property determines which of the type-specific properties are relevant. When evaluated on a cluster, the condition may match, not match, or fail to evaluate.
message	**string**	message provides additional information about the risk of updating, in the event that matchingRules match the cluster state. This is only to be consumed by humans. It may contain Line Feed characters (U+000A), which should be rendered as new lines.
name	**string**	name is the CamelCase reason for not recommending a conditional update, in the event that matchingRules match the cluster state.
url	**string**	url contains information about this risk.

6.1.15. .status.conditionalUpdates[].risks[].matchingRules

Description

matchingRules is a slice of conditions for deciding which clusters match the risk and which do not. The slice is ordered by decreasing precedence. The cluster-version operator will walk the slice in order, and stop after the first it can successfully evaluate. If no condition can be successfully evaluated, the update will not be recommended.

Type

array

6.1.16. .status.conditionalUpdates[].risks[].matchingRules[]

Description

ClusterCondition is a union of typed cluster conditions. The 'type' property determines which of the type-specific properties are relevant. When evaluated on a cluster, the condition may match, not match, or fail to evaluate.

Type

object

Required

- **type**

Property	Type	Description
promql	**object**	promQL represents a cluster condition based on PromQL.
type	**string**	type represents the cluster-condition type. This defines the members and semantics of any additional properties.

6.1.17. .status.conditionalUpdates[].risks[].matchingRules[].promql

Description

promQL represents a cluster condition based on PromQL.

Type

object

Required

- **promql**

Property	Type	Description
promql	**string**	PromQL is a PromQL query classifying clusters. This query query should return a 1 in the match case and a 0 in the does-not-match case. Queries which return no time series, or which return values besides 0 or 1, are evaluation failures.

6.1.18. .status.conditions

Description

conditions provides information about the cluster version. The condition "Available" is set to true if

the desiredUpdate has been reached. The condition "Progressing" is set to true if an update is being applied. The condition "Degraded" is set to true if an update is currently blocked by a temporary or permanent error. Conditions are only valid for the current desiredUpdate when metadata.generation is equal to status.generation.

Type

array

6.1.19. .status.conditions[]

Description

ClusterOperatorStatusCondition represents the state of the operator's managed and monitored components.

Type

object

Required

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

Property	Type	Description
lastTransitionTime	**string**	lastTransitionTime is the time of the last update to the current status property.
message	**string**	message provides additional information about the current condition. This is only to be consumed by humans. It may contain Line Feed characters (U+000A), which should be rendered as new lines.
reason	**string**	reason is the CamelCase reason for the condition's current status.
status	**string**	status of the condition, one of True, False, Unknown.
type	**string**	type specifies the aspect reported by this condition.

6.1.20. .status.desired

Description

desired is the version that the cluster is reconciling towards. If the cluster is not yet fully initialized desired will be set with the information available, which may be an image or a tag.

Type

object

Required

- **image**
- **version**

Property	Type	Description
channels	**array (string)**	channels is the set of Cincinnati channels to which the release currently belongs.
image	**string**	image is a container image location that contains the update. When this field is part of spec, image is optional if version is specified and the availableUpdates field contains a matching version.
url	**string**	url contains information about this release. This URL is set by the 'url' metadata property on a release or the metadata returned by the update API and should be displayed as a link in user interfaces. The URL field may not be set for test or nightly releases.
version	**string**	version is a semantic version identifying the update version. When this field is part of spec, version is optional if image is specified.

6.1.21. .status.history

Description

history contains a list of the most recent versions applied to the cluster. This value may be empty during cluster startup, and then will be updated when a new update is being applied. The newest update is first in the list and it is ordered by recency. Updates in the history have state Completed if the rollout completed - if an update was failing or halfway applied the state will be Partial. Only a limited amount of update history is preserved.

Type

array

6.1.22. .status.history[]

Description

UpdateHistory is a single attempted update to the cluster.

Type

object

Required

- **image**
- **startedTime**
- **state**
- **verified**

Property	Type	Description
acceptedRisks	**string**	acceptedRisks records risks which were accepted to initiate the update. For example, it may menition an Upgradeable=False or missing signature that was overriden via desiredUpdate.force, or an update that was initiated despite not being in the availableUpdates set of recommended update targets.
completionTime	``	completionTime, if set, is when the update was fully applied. The update that is currently being applied will have a null completion time. Completion time will always be set for entries that are not the current update (usually to the started time of the next update).
image	**string**	image is a container image location that contains the update. This value is always populated.
startedTime	**string**	startedTime is the time at which the update was started.

Property	Type	Description
state	**string**	state reflects whether the update was fully applied. The Partial state indicates the update is not fully applied, while the Completed state indicates the update was successfully rolled out at least once (all parts of the update successfully applied).
verified	**boolean**	verified indicates whether the provided update was properly verified before it was installed. If this is false the cluster may not be trusted. Verified does not cover upgradeable checks that depend on the cluster state at the time when the update target was accepted.
version	**string**	version is a semantic version identifying the update version. If the requested image does not define a version, or if a failure occurs retrieving the image, this value may be empty.

6.2. API ENDPOINTS

The following API endpoints are available:

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

- **PATCH**: partially update status of the specified ClusterVersion
- **PUT**: replace status of the specified ClusterVersion

6.2.1. /apis/config.openshift.io/v1/clusterversions

HTTP method

DELETE

Description

delete collection of ClusterVersion

Table 6.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind ClusterVersion

Table 6.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterVersionList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a ClusterVersion

Table 6.3. Query parameters

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

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

Table 6.4. Body parameters

Parameter	Type	Description
body	**ClusterVersion** schema	

Table 6.5. HTTP responses

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

6.2.2. /apis/config.openshift.io/v1/clusterversions/{name}

Table 6.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the ClusterVersion

HTTP method

DELETE

Description

delete a ClusterVersion

Table 6.7. Query parameters

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

Table 6.8. HTTP responses

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

HTTP method

GET

Description

read the specified ClusterVersion

Table 6.9. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterVersion** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ClusterVersion

Table 6.10. Query parameters

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

Table 6.11. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterVersion** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ClusterVersion

Table 6.12. Query parameters

Parameter	Type	Description

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

Table 6.13. Body parameters

Parameter	Type	Description
body	**ClusterVersion** schema	

Table 6.14. HTTP responses

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

6.2.3. /apis/config.openshift.io/v1/clusterversions/{name}/status

Table 6.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the ClusterVersion

HTTP method

GET

Description

read status of the specified ClusterVersion

Table 6.16. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterVersion** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ClusterVersion

Table 6.17. 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 6.18. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterVersion** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ClusterVersion

Table 6.19. 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: - 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 6.20. Body parameters

Parameter	Type	Description
body	**ClusterVersion** schema	

Table 6.21. HTTP responses

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

CHAPTER 7. CONSOLE [CONFIG.OPENSHIFT.IO/V1]

Description

Console holds cluster-wide configuration for the web console, including the logout URL, and reports the public URL of the console. The canonical name is **cluster**.

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

Type

object

Required

- **spec**

7.1. SPECIFICATION

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

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

7.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
authentication	**object**	ConsoleAuthentication defines a list of optional configuration for console authentication.

7.1.2. .spec.authentication

Description

ConsoleAuthentication defines a list of optional configuration for console authentication.

Type

object

Property	Type	Description
logoutRedirect	**string**	An optional, absolute URL to redirect web browsers to after logging out of the console. If not specified, it will redirect to the default login page. This is required when using an identity provider that supports single sign-on (SSO) such as: - OpenID (Keycloak, Azure) - RequestHeader (GSSAPI, SSPI, SAML) - OAuth (GitHub, GitLab, Google) Logging out of the console will destroy the user's token. The logoutRedirect provides the user the option to perform single logout (SLO) through the identity provider to destroy their single sign-on session.

7.1.3. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

Property	Type	Description
consoleURL	**string**	The URL for the console. This will be derived from the host for the route that is created for the console.

7.2. API ENDPOINTS

The following API endpoints are available:

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

7.2.1. /apis/config.openshift.io/v1/consoles

HTTP method

DELETE

Description

delete collection of Console

Table 7.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Console

Table 7.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ConsoleList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Console

Table 7.3. Query parameters

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

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

Table 7.4. Body parameters

Parameter	Type	Description
body	**Console** schema	

Table 7.5. HTTP responses

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

7.2.2. /apis/config.openshift.io/v1/consoles/{name}

Table 7.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Console

HTTP method

DELETE

Description

delete a Console

Table 7.7. Query parameters

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

Table 7.8. HTTP responses

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

HTTP method

GET

Description

read the specified Console

Table 7.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Console** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Console

Table 7.10. Query parameters

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

Table 7.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Console** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Console

Table 7.12. Query parameters

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

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

Table 7.13. Body parameters

Parameter	Type	Description
body	**Console** schema	

Table 7.14. HTTP responses

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

7.2.3. /apis/config.openshift.io/v1/consoles/{name}/status

Table 7.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Console

HTTP method

GET

Description

read status of the specified Console

Table 7.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Console** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Console

Table 7.17. Query parameters

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

Table 7.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Console** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Console

Table 7.19. Query parameters

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

Table 7.20. Body parameters

Parameter	Type	Description
body	**Console** schema	

Table 7.21. HTTP responses

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

CHAPTER 8. DNS [CONFIG.OPENSHIFT.IO/V1]

Description

DNS holds cluster-wide information about DNS. The canonical name is **cluster**

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

Type

object

Required

- **spec**

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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

8.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
baseDomain	**string**	baseDomain is the base domain of the cluster. All managed DNS records will be sub-domains of this base. For example, given the base domain **openshift.example.com**, an API server DNS record may be created for **cluster-api.openshift.example.com**. Once set, this field cannot be changed.
platform	**object**	platform holds configuration specific to the underlying infrastructure provider for DNS. When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time.
privateZone	**object**	privateZone is the location where all the DNS records that are only available internally to the cluster exist. If this field is nil, no private records should be created. Once set, this field cannot be changed.

Property	Type	Description
publicZone	**object**	publicZone is the location where all the DNS records that are publicly accessible to the internet exist. If this field is nil, no public records should be created. Once set, this field cannot be changed.

8.1.2. .spec.platform

Description

platform holds configuration specific to the underlying infrastructure provider for DNS. When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time.

Type

object

Required

- **type**

Property	Type	Description
aws	**object**	aws contains DNS configuration specific to the Amazon Web Services cloud provider.
type	**string**	type is the underlying infrastructure provider for the cluster. Allowed values: "", "AWS". Individual components may not support all platforms, and must handle unrecognized platforms with best-effort defaults.

8.1.3. .spec.platform.aws

Description

aws contains DNS configuration specific to the Amazon Web Services cloud provider.

Type

object

Property	Type	Description
privateZoneIAMRole	**string**	privateZoneIAMRole contains the ARN of an IAM role that should be assumed when performing operations on the cluster's private hosted zone specified in the cluster DNS config. When left empty, no role should be assumed.

8.1.4. .spec.privateZone

Description

privateZone is the location where all the DNS records that are only available internally to the cluster exist.
If this field is nil, no private records should be created.

Once set, this field cannot be changed.

Type

object

Property	Type	Description
id	**string**	id is the identifier that can be used to find the DNS hosted zone. on AWS zone can be fetched using **ID** as id in [1] on Azure zone can be fetched using **ID** as a pre-determined name in [2], on GCP zone can be fetched using **ID** as a pre-determined name in [3]. [1]: https://docs.aws.amazon.com/cli/latest/reference/route53/get-hosted-zone.html#options [2]: https://docs.microsoft.com/en-us/cli/azure/network/dns/zone?view=azure-cli-latest#az-network-dns-zone-show [3]: https://cloud.google.com/dns/docs/reference/v1/managedZones/get

Property	Type	Description
tags	**object (string)**	tags can be used to query the DNS hosted zone. on AWS, resourcegroupstaggingapi [1] can be used to fetch a zone using **Tags** as tag-filters, [1]: https://docs.aws.amazon.com/cli/latest/reference/resourcegroupstaggingapi/get-resources.html#options

8.1.5. .spec.publicZone

Description

publicZone is the location where all the DNS records that are publicly accessible to the internet exist. If this field is nil, no public records should be created.

Once set, this field cannot be changed.

Type

object

Property	Type	Description

Property	Type	Description
id	**string**	id is the identifier that can be used to find the DNS hosted zone. on AWS zone can be fetched using **ID** as id in [1] on Azure zone can be fetched using **ID** as a pre-determined name in [2], on GCP zone can be fetched using **ID** as a pre-determined name in [3]. [1]: https://docs.aws.amazon.com/cli/latest/reference/route53/get-hosted-zone.html#options [2]: https://docs.microsoft.com/en-us/cli/azure/network/dns/zone?view=azure-cli-latest#az-network-dns-zone-show [3]: https://cloud.google.com/dns/docs/reference/v1/managedZones/get
tags	**object (string)**	tags can be used to query the DNS hosted zone. on AWS, resourcegroupstaggingapi [1] can be used to fetch a zone using **Tags** as tag-filters, [1]: https://docs.aws.amazon.com/cli/latest/reference/resourcegroupstaggingapi/get-resources.html#options

8.1.6. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

8.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/dnses**
  - **DELETE**: delete collection of DNS

- **GET**: list objects of kind DNS
- **POST**: create a DNS

- **/apis/config.openshift.io/v1/dnses/{name}**
  - **DELETE**: delete a DNS
  - **GET**: read the specified DNS
  - **PATCH**: partially update the specified DNS
  - **PUT**: replace the specified DNS
- **/apis/config.openshift.io/v1/dnses/{name}/status**
  - **GET**: read status of the specified DNS
  - **PATCH**: partially update status of the specified DNS
  - **PUT**: replace status of the specified DNS

8.2.1. /apis/config.openshift.io/v1/dnses

HTTP method

DELETE

Description

delete collection of DNS

Table 8.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind DNS

Table 8.2. HTTP responses

HTTP code	Reponse body
200 - OK	**DNSList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a DNS

Table 8.3. Query parameters

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

Table 8.4. Body parameters

Parameter	Type	Description
body	**DNS** schema	

Table 8.5. HTTP responses

HTTP code	Reponse body
200 - OK	**DNS** schema
201 - Created	**DNS** schema
202 - Accepted	**DNS** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

8.2.2. /apis/config.openshift.io/v1/dnses/{name}

Table 8.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the DNS

HTTP method

DELETE

Description

delete a DNS

Table 8.7. 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 8.8. HTTP responses

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

HTTP method

GET

Description

read the specified DNS

Table 8.9. HTTP responses

HTTP code	Reponse body
200 - OK	**DNS** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified DNS

Table 8.10. 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 8.11. HTTP responses

HTTP code	Reponse body
200 - OK	**DNS** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified DNS

Table 8.12. Query parameters

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

Table 8.13. Body parameters

Parameter	Type	Description
body	**DNS** schema	

Table 8.14. HTTP responses

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

8.2.3. /apis/config.openshift.io/v1/dnses/{name}/status

Table 8.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the DNS

HTTP method

GET

Description

read status of the specified DNS

Table 8.16. HTTP responses

HTTP code	Reponse body
200 - OK	**DNS** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified DNS

Table 8.17. 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 8.18. HTTP responses

HTTP code	Reponse body
200 - OK	**DNS** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified DNS

Table 8.19. 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: - 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 8.20. Body parameters

Parameter	Type	Description
body	**DNS** schema	

Table 8.21. HTTP responses

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

CHAPTER 9. FEATUREGATE [CONFIG.OPENSHIFT.IO/V1]

Description

Feature holds cluster-wide information about feature gates. The canonical name is **cluster**

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

Type

object

Required

- **spec**

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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

9.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
customNoUpgrade	``	customNoUpgrade allows the enabling or disabling of any feature. Turning this feature set on IS NOT SUPPORTED, CANNOT BE UNDONE, and PREVENTS UPGRADES. Because of its nature, this setting cannot be validated. If you have any typos or accidentally apply invalid combinations your cluster may fail in an unrecoverable way. featureSet must equal "CustomNoUpgrade" must be set to use this field.
featureSet	**string**	featureSet changes the list of features in the cluster. The default is empty. Be very careful adjusting this setting. Turning on or off features may cause irreversible changes in your cluster which cannot be undone.

9.1.2. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

Property	Type	Description
conditions	**array**	conditions represent the observations of the current state. Known .status.conditions.type are: "DeterminationDegraded"
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.
featureGates	**array**	featureGates contains a list of enabled and disabled featureGates that are keyed by payloadVersion. Operators other than the CVO and cluster-config-operator, must read the .status.featureGates, locate the version they are managing, find the enabled/disabled featuregates and make the operand and operator match. The enabled/disabled values for a particular version may change during the life of the cluster as various .spec.featureSet values are selected. Operators may choose to restart their processes to pick up these changes, but remembering past enable/disable lists is beyond the scope of this API and is the responsibility of individual operators. Only featureGates with .version in the ClusterVersion.status will be present in this list.
featureGates[]	**object**	

9.1.3. .status.conditions

Description

conditions represent the observations of the current state. Known .status.conditions.type are: "DeterminationDegraded"

Type

array

9.1.4. .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.

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

9.1.5. .status.featureGates

Description

featureGates contains a list of enabled and disabled featureGates that are keyed by payloadVersion. Operators other than the CVO and cluster-config-operator, must read the .status.featureGates, locate the version they are managing, find the enabled/disabled featuregates and make the operand and operator match. The enabled/disabled values for a particular version may change during the life of the cluster as various .spec.featureSet values are selected. Operators may choose to restart their processes to pick up these changes, but remembering past enable/disable lists is beyond the scope of this API and is the responsibility of individual operators. Only featureGates with .version in the ClusterVersion.status will be present in this list.

Type

array

9.1.6. .status.featureGates[]

Description

Type

object

Required

- **version**

Property	Type	Description
disabled	**array**	disabled is a list of all feature gates that are disabled in the cluster for the named version.
disabled[]	**object**	
enabled	**array**	enabled is a list of all feature gates that are enabled in the cluster for the named version.
enabled[]	**object**	

Property	Type	Description
version	**string**	version matches the version provided by the ClusterVersion and in the ClusterOperator.Status.Versions field.

9.1.7. .status.featureGates[].disabled

Description

disabled is a list of all feature gates that are disabled in the cluster for the named version.

Type

array

9.1.8. .status.featureGates[].disabled[]

Description

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the name of the FeatureGate.

9.1.9. .status.featureGates[].enabled

Description

enabled is a list of all feature gates that are enabled in the cluster for the named version.

Type

array

9.1.10. .status.featureGates[].enabled[]

Description

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the name of the FeatureGate.

9.2. API ENDPOINTS

The following API endpoints are available:

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

9.2.1. /apis/config.openshift.io/v1/featuregates

HTTP method

DELETE

Description

delete collection of FeatureGate

Table 9.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind FeatureGate

Table 9.2. HTTP responses

HTTP code	Reponse body
200 - OK	**FeatureGateList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a FeatureGate

Table 9.3. Query parameters

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

Table 9.4. Body parameters

Parameter	Type	Description
body	**FeatureGate** schema	

Table 9.5. HTTP responses

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

9.2.2. /apis/config.openshift.io/v1/featuregates/{name}

Table 9.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the FeatureGate

HTTP method

DELETE

Description

delete a FeatureGate

Table 9.7. 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 9.8. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema

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

HTTP method

GET

Description

read the specified FeatureGate

Table 9.9. HTTP responses

HTTP code	Reponse body
200 - OK	**FeatureGate** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified FeatureGate

Table 9.10. Query parameters

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

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

Table 9.11. HTTP responses

HTTP code	Reponse body
200 - OK	**FeatureGate** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified FeatureGate

Table 9.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 9.13. Body parameters

Parameter	Type	Description
body	**FeatureGate** schema	

Table 9.14. HTTP responses

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

9.2.3. /apis/config.openshift.io/v1/featuregates/{name}/status

Table 9.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the FeatureGate

HTTP method

GET

Description

read status of the specified FeatureGate

Table 9.16. HTTP responses

HTTP code	Reponse body
200 - OK	**FeatureGate** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified FeatureGate

Table 9.17. Query parameters

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

Table 9.18. HTTP responses

HTTP code	Reponse body
200 - OK	**FeatureGate** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified FeatureGate

Table 9.19. Query parameters

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

Table 9.20. Body parameters

Parameter	Type	Description
body	**FeatureGate** schema	

Table 9.21. HTTP responses

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

CHAPTER 10. HELMCHARTREPOSITORY [HELM.OPENSHIFT.IO/V1BETA1]

Description

HelmChartRepository holds cluster-wide configuration for proxied Helm chart repository Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	Observed status of the repository within the cluster..

10.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
connectionConfig	**object**	Required configuration for connecting to the chart repo
description	**string**	Optional human readable repository description, it can be used by UI for displaying purposes
disabled	**boolean**	If set to true, disable the repo usage in the cluster/namespace
name	**string**	Optional associated human readable repository name, it can be used by UI for displaying purposes

10.1.2. .spec.connectionConfig

Description

Required configuration for connecting to the chart repo

Type

object

Property	Type	Description

Property	Type	Description
ca	**object**	ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca-bundle.crt" is used to locate the data. If empty, the default system roots are used. The namespace for this config map is openshift-config.
tlsClientConfig	**object**	tlsClientConfig is an optional reference to a secret by name that contains the PEM-encoded TLS client certificate and private key to present when connecting to the server. The key "tls.crt" is used to locate the client certificate. The key "tls.key" is used to locate the private key. The namespace for this secret is openshift-config.
url	**string**	Chart repository URL

10.1.3. .spec.connectionConfig.ca

Description

ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca-bundle.crt" is used to locate the data. If empty, the default system roots are used. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

10.1.4. .spec.connectionConfig.tlsClientConfig

Description

tlsClientConfig is an optional reference to a secret by name that contains the PEM-encoded TLS client certificate and private key to present when connecting to the server. The key "tls.crt" is used to locate the client certificate. The key "tls.key" is used to locate the private key. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

10.1.5. .status

Description

Observed status of the repository within the cluster..

Type

object

Property	Type	Description
conditions	**array**	conditions is a list of conditions and their statuses
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.

10.1.6. .status.conditions

Description

conditions is a list of conditions and their statuses

Type

array

10.1.7. .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.

Property	Type	Description
type	**string**	type of condition in CamelCase or in foo.example.com/CamelCase.

10.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/helm.openshift.io/v1beta1/helmchartrepositories**
  - **DELETE**: delete collection of HelmChartRepository
  - **GET**: list objects of kind HelmChartRepository
  - **POST**: create a HelmChartRepository
- **/apis/helm.openshift.io/v1beta1/helmchartrepositories/{name}**
  - **DELETE**: delete a HelmChartRepository
  - **GET**: read the specified HelmChartRepository
  - **PATCH**: partially update the specified HelmChartRepository
  - **PUT**: replace the specified HelmChartRepository
- **/apis/helm.openshift.io/v1beta1/helmchartrepositories/{name}/status**
  - **GET**: read status of the specified HelmChartRepository
  - **PATCH**: partially update status of the specified HelmChartRepository
  - **PUT**: replace status of the specified HelmChartRepository

10.2.1. /apis/helm.openshift.io/v1beta1/helmchartrepositories

HTTP method

DELETE

Description

delete collection of HelmChartRepository

Table 10.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind HelmChartRepository

Table 10.2. HTTP responses

HTTP code	Reponse body
200 - OK	**HelmChartRepositoryList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a HelmChartRepository

Table 10.3. Query parameters

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

Table 10.4. Body parameters

Parameter	Type	Description
body	**HelmChartRepository** schema	

Table 10.5. HTTP responses

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

10.2.2. /apis/helm.openshift.io/v1beta1/helmchartrepositories/{name}

Table 10.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the HelmChartRepository

HTTP method

DELETE

Description

delete a HelmChartRepository

Table 10.7. 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 10.8. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema

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

HTTP method

GET

Description

read the specified HelmChartRepository

Table 10.9. HTTP responses

HTTP code	Reponse body
200 - OK	**HelmChartRepository** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified HelmChartRepository

Table 10.10. Query parameters

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

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

Table 10.11. HTTP responses

HTTP code	Reponse body
200 - OK	**HelmChartRepository** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified HelmChartRepository

Table 10.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 10.13. Body parameters

Parameter	Type	Description
body	**HelmChartRepository** schema	

Table 10.14. HTTP responses

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

10.2.3. /apis/helm.openshift.io/v1beta1/helmchartrepositories/{name}/status

Table 10.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the HelmChartRepository

HTTP method

GET

Description

read status of the specified HelmChartRepository

Table 10.16. HTTP responses

HTTP code	Reponse body
200 - OK	**HelmChartRepository** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified HelmChartRepository

Table 10.17. Query parameters

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

Table 10.18. HTTP responses

HTTP code	Reponse body
200 - OK	**HelmChartRepository** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified HelmChartRepository

Table 10.19. Query parameters

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

Table 10.20. Body parameters

Parameter	Type	Description
body	**HelmChartRepository** schema	

Table 10.21. HTTP responses

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

CHAPTER 11. IMAGE [CONFIG.OPENSHIFT.IO/V1]

Description

Image governs policies related to imagestream imports and runtime configuration for external registries. It allows cluster admins to configure which registries OpenShift is allowed to import images from, extra CA trust bundles for external registries, and policies to block or allow registry hostnames. When exposing OpenShift's image registry to the public, this also lets cluster admins specify the external hostname.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

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

Property	Type	Description
spec	**object**	spec holds user settable values for configuration
status	**object**	status holds observed values from the cluster. They may not be overridden.

11.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
additionalTrustedCA	**object**	additionalTrustedCA is a reference to a ConfigMap containing additional CAs that should be trusted during imagestream import, pod image pull, build image pull, and imageregistry pullthrough. The namespace for this config map is openshift-config.
allowedRegistriesForImport	**array**	allowedRegistriesForImport limits the container image registries that normal users may import images from. Set this list to the registries that you trust to contain valid Docker images and that you want applications to be able to import from. Users with permission to create Images or ImageStreamMappings via the API are not affected by this policy - typically only administrators or system integrations will have those permissions.
allowedRegistriesForImport[]	**object**	RegistryLocation contains a location of the registry specified by the registry domain name. The domain name might include wildcards, like '*' or '??'.

Property	Type	Description
externalRegistryHostnames	**array (string)**	externalRegistryHostnames provides the hostnames for the default external image registry. The external hostname should be set only when the image registry is exposed externally. The first value is used in 'publicDockerImageRepository' field in ImageStreams. The value must be in "hostname[:port]" format.
registrySources	**object**	registrySources contains configuration that determines how the container runtime should treat individual registries when accessing images for builds+pods. (e.g. whether or not to allow insecure access). It does not contain configuration for the internal cluster registry.

11.1.2. .spec.additionalTrustedCA

Description

additionalTrustedCA is a reference to a ConfigMap containing additional CAs that should be trusted during imagestream import, pod image pull, build image pull, and imageregistry pullthrough. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

11.1.3. .spec.allowedRegistriesForImport

Description

allowedRegistriesForImport limits the container image registries that normal users may import images from. Set this list to the registries that you trust to contain valid Docker images and that you want applications to be able to import from. Users with permission to create Images or

ImageStreamMappings via the API are not affected by this policy - typically only administrators or system integrations will have those permissions.

Type

array

11.1.4. .spec.allowedRegistriesForImport[]

Description

RegistryLocation contains a location of the registry specified by the registry domain name. The domain name might include wildcards, like '*' or '??'.

Type

object

Property	Type	Description
domainName	**string**	domainName specifies a domain name for the registry In case the registry use non-standard (80 or 443) port, the port should be included in the domain name as well.
insecure	**boolean**	insecure indicates whether the registry is secure (https) or insecure (http) By default (if not specified) the registry is assumed as secure.

11.1.5. .spec.registrySources

Description

registrySources contains configuration that determines how the container runtime should treat individual registries when accessing images for builds+pods. (e.g. whether or not to allow insecure access). It does not contain configuration for the internal cluster registry.

Type

object

Property	Type	Description
allowedRegistries	**array (string)**	allowedRegistries are the only registries permitted for image pull and push actions. All other registries are denied. Only one of BlockedRegistries or AllowedRegistries may be set.

Property	Type	Description
blockedRegistries	**array (string)**	blockedRegistries cannot be used for image pull and push actions. All other registries are permitted. Only one of BlockedRegistries or AllowedRegistries may be set.
containerRuntimeSearchRegistries	**array (string)**	containerRuntimeSearchRegistries are registries that will be searched when pulling images that do not have fully qualified domains in their pull specs. Registries will be searched in the order provided in the list. Note: this search list only works with the container runtime, i.e CRI-O. Will NOT work with builds or imagestream imports.
insecureRegistries	**array (string)**	insecureRegistries are registries which do not have a valid TLS certificates or only support HTTP connections.

11.1.6. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

Property	Type	Description
externalRegistryHostnames	**array (string)**	externalRegistryHostnames provides the hostnames for the default external image registry. The external hostname should be set only when the image registry is exposed externally. The first value is used in 'publicDockerImageRepository' field in ImageStreams. The value must be in "hostname[:port]" format.

Property	Type	Description
internalRegistryHostname	**string**	internalRegistryHostname sets the hostname for the default internal image registry. The value must be in "hostname[:port]" format. This value is set by the image registry operator which controls the internal registry hostname.

11.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/images**
  - **DELETE**: delete collection of Image
  - **GET**: list objects of kind Image
  - **POST**: create an Image
- **/apis/config.openshift.io/v1/images/{name}**
  - **DELETE**: delete an Image
  - **GET**: read the specified Image
  - **PATCH**: partially update the specified Image
  - **PUT**: replace the specified Image
- **/apis/config.openshift.io/v1/images/{name}/status**
  - **GET**: read status of the specified Image
  - **PATCH**: partially update status of the specified Image
  - **PUT**: replace status of the specified Image

11.2.1. /apis/config.openshift.io/v1/images

HTTP method

DELETE

Description

delete collection of Image

Table 11.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Image

Table 11.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an Image

Table 11.3. Query parameters

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

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

Table 11.4. Body parameters

Parameter	Type	Description
body	**Image** schema	

Table 11.5. HTTP responses

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

11.2.2. /apis/config.openshift.io/v1/images/{name}

Table 11.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Image

HTTP method

DELETE

Description

delete an Image

Table 11.7. 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 11.8. HTTP responses

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

HTTP method

GET

Description

read the specified Image

Table 11.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Image** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Image

Table 11.10. 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 11.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Image** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Image

Table 11.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 11.13. Body parameters

Parameter	Type	Description
body	**Image** schema	

Table 11.14. HTTP responses

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

11.2.3. /apis/config.openshift.io/v1/images/{name}/status

Table 11.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Image

HTTP method

GET

Description

read status of the specified Image

Table 11.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Image** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Image

Table 11.17. Query parameters

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

Table 11.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Image** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Image

Table 11.19. Query parameters

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

Table 11.20. Body parameters

Parameter	Type	Description
body	**Image** schema	

Table 11.21. HTTP responses

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

CHAPTER 12. IMAGEDIGESTMIRRORSET [CONFIG.OPENSHIFT.IO/V1]

Description

ImageDigestMirrorSet holds cluster-wide information about how to handle registry mirror rules on using digest pull specification. When multiple policies are defined, the outcome of the behavior is defined on each field.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

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

Property	Type	Description
spec	**object**	spec holds user settable values for configuration
status	**object**	status contains the observed state of the resource.

12.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
imageDigestMirrors	**array**	imageDigestMirrors allows images referenced by image digests in pods to be pulled from alternative mirrored repository locations. The image pull specification provided to the pod will be compared to the source locations described in imageDigestMirrors and the image may be pulled down from any of the mirrors in the list instead of the specified repository allowing administrators to choose a potentially faster mirror. To use mirrors to pull images using tag specification, users should configure a list of mirrors using "ImageTagMirrorSet" CRD. If the image pull specification matches the repository of "source" in multiple imagedigestmirrorset objects, only the objects which define the most specific namespace match will be used. For example, if there are objects using quay.io/libpod and quay.io/libpod/busybox as the "source", only the objects using quay.io/libpod/busybox are going to apply for pull specification quay.io/libpod/busybox. Each “source” repository is treated

Property	Type	Description
		independently; configurations for different “source” repositories don’t interact. If the "mirrors" is not specified, the image will continue to be pulled from the specified repository in the pull spec. When multiple policies are defined for the same “source” repository, the sets of defined mirrors will be merged together, preserving the relative order of the mirrors, if possible. For example, if policy A has mirrors **a, b, c** and policy B has mirrors **c, d, e**, the mirrors will be used in the order **a, b, c, d, e**. If the orders of mirror entries conflict (e.g. **a, b** vs. **b, a**) the configuration is not rejected but the resulting order is unspecified. Users who want to use a specific order of mirrors, should configure them into one list of mirrors using the expected order.
imageDigestMirrors[]	**object**	ImageDigestMirrors holds cluster-wide information about how to handle mirrors in the registries config.

12.1.2. .spec.imageDigestMirrors

Description

imageDigestMirrors allows images referenced by image digests in pods to be pulled from alternative mirrored repository locations. The image pull specification provided to the pod will be compared to the source locations described in imageDigestMirrors and the image may be pulled down from any of the mirrors in the list instead of the specified repository allowing administrators to choose a potentially faster mirror. To use mirrors to pull images using tag specification, users should configure a list of mirrors using "ImageTagMirrorSet" CRD.
If the image pull specification matches the repository of "source" in multiple imagedigestmirrorset objects, only the objects which define the most specific namespace match will be used. For example, if there are objects using quay.io/libpod and quay.io/libpod/busybox as the "source", only the objects using quay.io/libpod/busybox are going to apply for pull specification quay.io/libpod/busybox. Each “source” repository is treated independently; configurations for different “source” repositories don’t interact.

If the "mirrors" is not specified, the image will continue to be pulled from the specified repository in the pull spec.

When multiple policies are defined for the same “source” repository, the sets of defined mirrors will be merged together, preserving the relative order of the mirrors, if possible. For example, if policy A

has mirrors **a, b, c** and policy B has mirrors **c, d, e**, the mirrors will be used in the order **a, b, c, d, e**. If the orders of mirror entries conflict (e.g. **a, b** vs. **b, a**) the configuration is not rejected but the resulting order is unspecified. Users who want to use a specific order of mirrors, should configure them into one list of mirrors using the expected order.

Type

array

12.1.3. .spec.imageDigestMirrors[]

Description

ImageDigestMirrors holds cluster-wide information about how to handle mirrors in the registries config.

Type

object

Required

- **source**

Property	Type	Description
mirrorSourcePolicy	**string**	mirrorSourcePolicy defines the fallback policy if fails to pull image from the mirrors. If unset, the image will continue to be pulled from the the repository in the pull spec. sourcePolicy is valid configuration only when one or more mirrors are in the mirror list.

Property	Type	Description
mirrors	**array (string)**	mirrors is zero or more locations that may also contain the same images. No mirror will be configured if not specified. Images can be pulled from these mirrors only if they are referenced by their digests. The mirrored location is obtained by replacing the part of the input reference that matches source by the mirrors entry, e.g. for registry.redhat.io/product/repo reference, a (source, mirror) pair *.redhat.io, mirror.local/redhat causes a mirror.local/redhat/product/repo repository to be used. The order of mirrors in this list is treated as the user's desired priority, while source is by default considered lower priority than all mirrors. If no mirror is specified or all image pulls from the mirror list fail, the image will continue to be pulled from the repository in the pull spec unless explicitly prohibited by "mirrorSourcePolicy" Other cluster configuration, including (but not limited to) other imageDigestMirrors objects, may impact the exact order mirrors are contacted in, or some mirrors may be contacted in parallel, so this should be considered a preference rather than a guarantee of ordering. "mirrors" uses one of the following formats: host[:port] host[:port]/namespace[/namespace…] host[:port]/namespace[/namespace…]/repo for more information about the format, see the document about the location field: https://github.com/containers/image/blob/main/docs/containers-registries.conf.5.md#choosing-a-registry-toml-table

Property	Type	Description
source	**string**	source matches the repository that users refer to, e.g. in image pull specifications. Setting source to a registry hostname e.g. docker.io. quay.io, or registry.redhat.io, will match the image pull specification of corressponding registry. "source" uses one of the following formats: host[:port] host[:port]/namespace[/namespace...] host[:port]/namespace[/namespace...]/repo [*.]host for more information about the format, see the document about the location field: https://github.com/containers/image/blob/main/docs/containers-registries.conf.5.md#choosing-a-registry-toml-table

12.1.4. .status

Description

status contains the observed state of the resource.

Type

object

12.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/imagedigestmirrorsets**
  - **DELETE**: delete collection of ImageDigestMirrorSet
  - **GET**: list objects of kind ImageDigestMirrorSet
  - **POST**: create an ImageDigestMirrorSet
- **/apis/config.openshift.io/v1/imagedigestmirrorsets/{name}**
  - **DELETE**: delete an ImageDigestMirrorSet
  - **GET**: read the specified ImageDigestMirrorSet
  - **PATCH**: partially update the specified ImageDigestMirrorSet
  - **PUT**: replace the specified ImageDigestMirrorSet

- **/apis/config.openshift.io/v1/imagedigestmirrorsets/{name}/status**
  - **GET**: read status of the specified ImageDigestMirrorSet
  - **PATCH**: partially update status of the specified ImageDigestMirrorSet
  - **PUT**: replace status of the specified ImageDigestMirrorSet

12.2.1. /apis/config.openshift.io/v1/imagedigestmirrorsets

HTTP method

DELETE

Description

delete collection of ImageDigestMirrorSet

Table 12.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind ImageDigestMirrorSet

Table 12.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageDigestMirrorSetList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an ImageDigestMirrorSet

Table 12.3. Query parameters

Parameter	Type	Description

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

Table 12.4. Body parameters

Parameter	Type	Description
body	**ImageDigestMirrorSet** schema	

Table 12.5. HTTP responses

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

12.2.2. /apis/config.openshift.io/v1/imagedigestmirrorsets/{name}

Table 12.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the ImageDigestMirrorSet

HTTP method

DELETE

Description

delete an ImageDigestMirrorSet

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 ImageDigestMirrorSet

Table 12.9. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageDigestMirrorSet** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ImageDigestMirrorSet

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	Reponse body
200 - OK	**ImageDigestMirrorSet** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ImageDigestMirrorSet

Table 12.12. Query parameters

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

Table 12.13. Body parameters

Parameter	Type	Description
body	**ImageDigestMirrorSet** schema	

Table 12.14. HTTP responses

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

12.2.3. /apis/config.openshift.io/v1/imagedigestmirrorsets/{name}/status

Table 12.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the ImageDigestMirrorSet

HTTP method

GET

Description

read status of the specified ImageDigestMirrorSet

Table 12.16. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageDigestMirrorSet** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ImageDigestMirrorSet

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	Reponse body
200 - OK	**ImageDigestMirrorSet** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ImageDigestMirrorSet

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: - 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	**ImageDigestMirrorSet** schema	

Table 12.21. HTTP responses

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

CHAPTER 13. IMAGECONTENTPOLICY [CONFIG.OPENSHIFT.IO/V1]

Description

ImageContentPolicy holds cluster-wide information about how to handle registry mirror rules. When multiple policies are defined, the outcome of the behavior is defined on each field.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

13.1. 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

Property	Type	Description
spec	**object**	spec holds user settable values for configuration

13.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description

Property	Type	Description
repositoryDigestMirrors	**array**	repositoryDigestMirrors allows images referenced by image digests in pods to be pulled from alternative mirrored repository locations. The image pull specification provided to the pod will be compared to the source locations described in RepositoryDigestMirrors and the image may be pulled down from any of the mirrors in the list instead of the specified repository allowing administrators to choose a potentially faster mirror. To pull image from mirrors by tags, should set the "allowMirrorByTags". Each “source” repository is treated independently; configurations for different “source” repositories don’t interact. If the "mirrors" is not specified, the image will continue to be pulled from the specified repository in the pull spec. When multiple policies are defined for the same “source” repository, the sets of defined mirrors will be merged together, preserving the relative order of the mirrors, if possible. For example, if policy A has mirrors **a, b, c** and policy B has mirrors **c, d, e**, the mirrors will be used in the order **a, b, c, d, e**. If the orders of mirror entries conflict (e.g. **a, b** vs. **b, a**) the configuration is not rejected but the resulting order is unspecified.
repositoryDigestMirrors[]	**object**	RepositoryDigestMirrors holds cluster-wide information about how to handle mirrors in the registries config.

13.1.2. .spec.repositoryDigestMirrors

Description

repositoryDigestMirrors allows images referenced by image digests in pods to be pulled from alternative mirrored repository locations. The image pull specification provided to the pod will be compared to the source locations described in RepositoryDigestMirrors and the image may be pulled down from any of the mirrors in the list instead of the specified repository allowing administrators to choose a potentially faster mirror. To pull image from mirrors by tags, should set the "allowMirrorByTags".
Each “source” repository is treated independently; configurations for different “source” repositories don’t interact.

If the "mirrors" is not specified, the image will continue to be pulled from the specified repository in the pull spec.

When multiple policies are defined for the same “source” repository, the sets of defined mirrors will be merged together, preserving the relative order of the mirrors, if possible. For example, if policy A has mirrors **a, b, c** and policy B has mirrors **c, d, e**, the mirrors will be used in the order **a, b, c, d, e**. If the orders of mirror entries conflict (e.g. **a, b** vs. **b, a**) the configuration is not rejected but the resulting order is unspecified.

Type

array

13.1.3. .spec.repositoryDigestMirrors[]

Description

RepositoryDigestMirrors holds cluster-wide information about how to handle mirrors in the registries config.

Type

object

Required

- **source**

Property	Type	Description
allowMirrorByTags	**boolean**	allowMirrorByTags if true, the mirrors can be used to pull the images that are referenced by their tags. Default is false, the mirrors only work when pulling the images that are referenced by their digests. Pulling images by tag can potentially yield different images, depending on which endpoint we pull from. Forcing digest-pulls for mirrors avoids that issue.

Property	Type	Description
mirrors	**array (string)**	mirrors is zero or more repositories that may also contain the same images. If the "mirrors" is not specified, the image will continue to be pulled from the specified repository in the pull spec. No mirror will be configured. The order of mirrors in this list is treated as the user's desired priority, while source is by default considered lower priority than all mirrors. Other cluster configuration, including (but not limited to) other repositoryDigestMirrors objects, may impact the exact order mirrors are contacted in, or some mirrors may be contacted in parallel, so this should be considered a preference rather than a guarantee of ordering.
source	**string**	source is the repository that users refer to, e.g. in image pull specifications.

13.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/imagecontentpolicies**
  - **DELETE**: delete collection of ImageContentPolicy
  - **GET**: list objects of kind ImageContentPolicy
  - **POST**: create an ImageContentPolicy
- **/apis/config.openshift.io/v1/imagecontentpolicies/{name}**
  - **DELETE**: delete an ImageContentPolicy
  - **GET**: read the specified ImageContentPolicy
  - **PATCH**: partially update the specified ImageContentPolicy
  - **PUT**: replace the specified ImageContentPolicy
- **/apis/config.openshift.io/v1/imagecontentpolicies/{name}/status**
  - **GET**: read status of the specified ImageContentPolicy

- **PATCH**: partially update status of the specified ImageContentPolicy
- **PUT**: replace status of the specified ImageContentPolicy

13.2.1. /apis/config.openshift.io/v1/imagecontentpolicies

HTTP method

DELETE

Description

delete collection of ImageContentPolicy

Table 13.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind ImageContentPolicy

Table 13.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageContentPolicyList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an ImageContentPolicy

Table 13.3. Query parameters

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

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

Table 13.4. Body parameters

Parameter	Type	Description
body	**ImageContentPolicy** schema	

Table 13.5. HTTP responses

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

13.2.2. /apis/config.openshift.io/v1/imagecontentpolicies/{name}

Table 13.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the ImageContentPolicy

HTTP method

DELETE

Description

delete an ImageContentPolicy

Table 13.7. 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 13.8. HTTP responses

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

HTTP method

GET

Description

read the specified ImageContentPolicy

Table 13.9. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageContentPolicy** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ImageContentPolicy

Table 13.10. 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 13.11. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageContentPolicy** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ImageContentPolicy

Table 13.12. 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 13.13. Body parameters

Parameter	Type	Description
body	**ImageContentPolicy** schema	

Table 13.14. HTTP responses

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

13.2.3. /apis/config.openshift.io/v1/imagecontentpolicies/{name}/status

Table 13.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the ImageContentPolicy

HTTP method

GET

Description

read status of the specified ImageContentPolicy

Table 13.16. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageContentPolicy** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ImageContentPolicy

Table 13.17. 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 13.18. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageContentPolicy** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ImageContentPolicy

Table 13.19. 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: - 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 13.20. Body parameters

Parameter	Type	Description
body	**ImageContentPolicy** schema	

Table 13.21. HTTP responses

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

CHAPTER 14. IMAGETAGMIRRORSET [CONFIG.OPENSHIFT.IO/V1]

Description

ImageTagMirrorSet holds cluster-wide information about how to handle registry mirror rules on using tag pull specification. When multiple policies are defined, the outcome of the behavior is defined on each field.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

14.1. 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

Property	Type	Description
spec	**object**	spec holds user settable values for configuration
status	**object**	status contains the observed state of the resource.

14.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
imageTagMirrors	**array**	imageTagMirrors allows images referenced by image tags in pods to be pulled from alternative mirrored repository locations. The image pull specification provided to the pod will be compared to the source locations described in imageTagMirrors and the image may be pulled down from any of the mirrors in the list instead of the specified repository allowing administrators to choose a potentially faster mirror. To use mirrors to pull images using digest specification only, users should configure a list of mirrors using "ImageDigestMirrorSet" CRD. If the image pull specification matches the repository of "source" in multiple imagetagmirrorset objects, only the objects which define the most specific namespace match will be used. For example, if there are objects using quay.io/libpod and quay.io/libpod/busybox as the "source", only the objects using quay.io/libpod/busybox are going to apply for pull specification quay.io/libpod/busybox. Each “source” repository is treated

Property	Type	Description
		independently; configurations for different "source" repositories don't interact. If the "mirrors" is not specified, the image will continue to be pulled from the specified repository in the pull spec. When multiple policies are defined for the same "source" repository, the sets of defined mirrors will be merged together, preserving the relative order of the mirrors, if possible. For example, if policy A has mirrors **a, b, c** and policy B has mirrors **c, d, e**, the mirrors will be used in the order **a, b, c, d, e**. If the orders of mirror entries conflict (e.g. **a, b** vs. **b, a**) the configuration is not rejected but the resulting order is unspecified. Users who want to use a deterministic order of mirrors, should configure them into one list of mirrors using the expected order.
imageTagMirrors[]	**object**	ImageTagMirrors holds cluster-wide information about how to handle mirrors in the registries config.

14.1.2. .spec.imageTagMirrors

Description

imageTagMirrors allows images referenced by image tags in pods to be pulled from alternative mirrored repository locations. The image pull specification provided to the pod will be compared to the source locations described in imageTagMirrors and the image may be pulled down from any of the mirrors in the list instead of the specified repository allowing administrators to choose a potentially faster mirror. To use mirrors to pull images using digest specification only, users should configure a list of mirrors using "ImageDigestMirrorSet" CRD.
If the image pull specification matches the repository of "source" in multiple imagetagmirrorset objects, only the objects which define the most specific namespace match will be used. For example, if there are objects using quay.io/libpod and quay.io/libpod/busybox as the "source", only the objects using quay.io/libpod/busybox are going to apply for pull specification quay.io/libpod/busybox. Each "source" repository is treated independently; configurations for different "source" repositories don't interact.

If the "mirrors" is not specified, the image will continue to be pulled from the specified repository in the pull spec.

When multiple policies are defined for the same "source" repository, the sets of defined mirrors will be merged together, preserving the relative order of the mirrors, if possible. For example, if policy A

has mirrors **a, b, c** and policy B has mirrors **c, d, e**, the mirrors will be used in the order **a, b, c, d, e**. If the orders of mirror entries conflict (e.g. **a, b** vs. **b, a**) the configuration is not rejected but the resulting order is unspecified. Users who want to use a deterministic order of mirrors, should configure them into one list of mirrors using the expected order.

Type

array

14.1.3. .spec.imageTagMirrors[]

Description

ImageTagMirrors holds cluster-wide information about how to handle mirrors in the registries config.

Type

object

Required

- **source**

Property	Type	Description
mirrorSourcePolicy	**string**	mirrorSourcePolicy defines the fallback policy if fails to pull image from the mirrors. If unset, the image will continue to be pulled from the repository in the pull spec. sourcePolicy is valid configuration only when one or more mirrors are in the mirror list.

Property	Type	Description
mirrors	**array (string)**	mirrors is zero or more locations that may also contain the same images. No mirror will be configured if not specified. Images can be pulled from these mirrors only if they are referenced by their tags. The mirrored location is obtained by replacing the part of the input reference that matches source by the mirrors entry, e.g. for registry.redhat.io/product/repo reference, a (source, mirror) pair *.redhat.io, mirror.local/redhat causes a mirror.local/redhat/product/repo repository to be used. Pulling images by tag can potentially yield different images, depending on which endpoint we pull from. Configuring a list of mirrors using "ImageDigestMirrorSet" CRD and forcing digest-pulls for mirrors avoids that issue. The order of mirrors in this list is treated as the user's desired priority, while source is by default considered lower priority than all mirrors. If no mirror is specified or all image pulls from the mirror list fail, the image will continue to be pulled from the repository in the pull spec unless explicitly prohibited by "mirrorSourcePolicy". Other cluster configuration, including (but not limited to) other imageTagMirrors objects, may impact the exact order mirrors are contacted in, or some mirrors may be contacted in parallel, so this should be considered a preference rather than a guarantee of ordering. "mirrors" uses one of the following formats: host[:port] host[:port]/namespace[/namespace...] host[:port]/namespace[/namespace...]/repo for more information about the format, see the document about the location field: https://github.com/containers/im

Property	Type	Description
source	**string**	age/blob/main/docs/containers-registries.conf.5.md#choosing-a-registry-toml-table source matches the repository that users refer to, e.g. in image pull specifications. Setting source to a registry hostname e.g. docker.io. quay.io, or registry.redhat.io, will match the image pull specification of corressponding registry. "source" uses one of the following formats: host[:port] host[:port]/namespace[/namespace...] host[:port]/namespace[/namespace...]/repo [*.]host for more information about the format, see the document about the location field: https://github.com/containers/image/blob/main/docs/containers-registries.conf.5.md#choosing-a-registry-toml-table

14.1.4. .status

Description

status contains the observed state of the resource.

Type

object

14.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/imagetagmirrorsets**
  - **DELETE**: delete collection of ImageTagMirrorSet
  - **GET**: list objects of kind ImageTagMirrorSet
  - **POST**: create an ImageTagMirrorSet
- **/apis/config.openshift.io/v1/imagetagmirrorsets/{name}**
  - **DELETE**: delete an ImageTagMirrorSet
  - **GET**: read the specified ImageTagMirrorSet
  - **PATCH**: partially update the specified ImageTagMirrorSet
  - **PUT**: replace the specified ImageTagMirrorSet

- **/apis/config.openshift.io/v1/imagetagmirrorsets/{name}/status**
  - **GET**: read status of the specified ImageTagMirrorSet
  - **PATCH**: partially update status of the specified ImageTagMirrorSet
  - **PUT**: replace status of the specified ImageTagMirrorSet

14.2.1. /apis/config.openshift.io/v1/imagetagmirrorsets

HTTP method

DELETE

Description

delete collection of ImageTagMirrorSet

Table 14.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind ImageTagMirrorSet

Table 14.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageTagMirrorSetList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an ImageTagMirrorSet

Table 14.3. Query parameters

Parameter	Type	Description

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

Table 14.4. Body parameters

Parameter	Type	Description
body	**ImageTagMirrorSet** schema	

Table 14.5. HTTP responses

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

14.2.2. /apis/config.openshift.io/v1/imagetagmirrorsets/{name}

Table 14.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the ImageTagMirrorSet

HTTP method

DELETE

Description

delete an ImageTagMirrorSet

Table 14.7. 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 14.8. HTTP responses

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

HTTP method

GET

Description

read the specified ImageTagMirrorSet

Table 14.9. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageTagMirrorSet** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ImageTagMirrorSet

Table 14.10. 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 14.11. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageTagMirrorSet** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ImageTagMirrorSet

Table 14.12. Query parameters

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

Table 14.13. Body parameters

Parameter	Type	Description
body	**ImageTagMirrorSet** schema	

Table 14.14. HTTP responses

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

14.2.3. /apis/config.openshift.io/v1/imagetagmirrorsets/{name}/status

Table 14.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the ImageTagMirrorSet

HTTP method

GET

Description

read status of the specified ImageTagMirrorSet

Table 14.16. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageTagMirrorSet** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ImageTagMirrorSet

Table 14.17. 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 14.18. HTTP responses

HTTP code	Reponse body
200 - OK	**ImageTagMirrorSet** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ImageTagMirrorSet

Table 14.19. 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: - 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 14.20. Body parameters

Parameter	Type	Description
body	**ImageTagMirrorSet** schema	

Table 14.21. HTTP responses

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

CHAPTER 15. INFRASTRUCTURE [CONFIG.OPENSHIFT.IO/V1]

Description

Infrastructure holds cluster-wide information about Infrastructure. The canonical name is **cluster** Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

15.1. 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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

15.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
cloudConfig	**object**	cloudConfig is a reference to a ConfigMap containing the cloud provider configuration file. This configuration file is used to configure the Kubernetes cloud provider integration when using the built-in cloud provider integration or the external cloud controller manager. The namespace for this config map is openshift-config. cloudConfig should only be consumed by the kube_cloud_config controller. The controller is responsible for using the user configuration in the spec for various platforms and combining that with the user provided ConfigMap in this field to create a stitched kube cloud config. The controller generates a ConfigMap **kube-cloud-config** in **openshift-config-managed** namespace with the kube cloud config is stored in **cloud.conf** key. All the clients are expected to use the generated ConfigMap only.
platformSpec	**object**	platformSpec holds desired information specific to the underlying infrastructure provider.

15.1.2. .spec.cloudConfig

Description

cloudConfig is a reference to a ConfigMap containing the cloud provider configuration file. This configuration file is used to configure the Kubernetes cloud provider integration when using the built-in cloud provider integration or the external cloud controller manager. The namespace for this config map is openshift-config.
cloudConfig should only be consumed by the kube_cloud_config controller. The controller is responsible for using the user configuration in the spec for various platforms and combining that with the user provided ConfigMap in this field to create a stitched kube cloud config. The controller generates a ConfigMap **kube-cloud-config** in **openshift-config-managed** namespace with the kube cloud config is stored in **cloud.conf** key. All the clients are expected to use the generated ConfigMap only.

Type

object

Property	Type	Description
key	**string**	Key allows pointing to a specific key/value inside of the configmap. This is useful for logical file references.
name	**string**	

15.1.3. .spec.platformSpec

Description

platformSpec holds desired information specific to the underlying infrastructure provider.

Type

object

Property	Type	Description
alibabaCloud	**object**	AlibabaCloud contains settings specific to the Alibaba Cloud infrastructure provider.
aws	**object**	AWS contains settings specific to the Amazon Web Services infrastructure provider.
azure	**object**	Azure contains settings specific to the Azure infrastructure provider.

Property	Type	Description
baremetal	**object**	BareMetal contains settings specific to the BareMetal platform.
equinixMetal	**object**	EquinixMetal contains settings specific to the Equinix Metal infrastructure provider.
external	**object**	ExternalPlatformType represents generic infrastructure provider. Platform-specific components should be supplemented separately.
gcp	**object**	GCP contains settings specific to the Google Cloud Platform infrastructure provider.
ibmcloud	**object**	IBMCloud contains settings specific to the IBMCloud infrastructure provider.
kubevirt	**object**	Kubevirt contains settings specific to the kubevirt infrastructure provider.
nutanix	**object**	Nutanix contains settings specific to the Nutanix infrastructure provider.
openstack	**object**	OpenStack contains settings specific to the OpenStack infrastructure provider.
ovirt	**object**	Ovirt contains settings specific to the oVirt infrastructure provider.
powervs	**object**	PowerVS contains settings specific to the IBM Power Systems Virtual Servers infrastructure provider.

Property	Type	Description
type	**string**	type is the underlying infrastructure provider for the cluster. This value controls whether infrastructure automation such as service load balancers, dynamic volume provisioning, machine creation and deletion, and other integrations are enabled. If None, no infrastructure automation is enabled. Allowed values are "AWS", "Azure", "BareMetal", "GCP", "Libvirt", "OpenStack", "VSphere", "oVirt", "KubeVirt", "EquinixMetal", "PowerVS", "AlibabaCloud", "Nutanix" and "None". Individual components may not support all platforms, and must handle unrecognized platforms as None if they do not support that platform.
vsphere	**object**	VSphere contains settings specific to the VSphere infrastructure provider.

15.1.4. .spec.platformSpec.alibabaCloud

Description

AlibabaCloud contains settings specific to the Alibaba Cloud infrastructure provider.

Type

object

15.1.5. .spec.platformSpec.aws

Description

AWS contains settings specific to the Amazon Web Services infrastructure provider.

Type

object

Property	Type	Description

Property	Type	Description
serviceEndpoints	**array**	serviceEndpoints list contains custom endpoints which will override default service endpoint of AWS Services. There must be only one ServiceEndpoint for a service.
serviceEndpoints[]	**object**	AWSServiceEndpoint store the configuration of a custom url to override existing defaults of AWS Services.

15.1.6. .spec.platformSpec.aws.serviceEndpoints

Description

serviceEndpoints list contains custom endpoints which will override default service endpoint of AWS Services. There must be only one ServiceEndpoint for a service.

Type

array

15.1.7. .spec.platformSpec.aws.serviceEndpoints[]

Description

AWSServiceEndpoint store the configuration of a custom url to override existing defaults of AWS Services.

Type

object

Property	Type	Description
name	**string**	name is the name of the AWS service. The list of all the service names can be found at https://docs.aws.amazon.com/general/latest/gr/aws-service-information.html This must be provided and cannot be empty.
url	**string**	url is fully qualified URI with scheme https, that overrides the default generated endpoint for a client. This must be provided and cannot be empty.

15.1.8. .spec.platformSpec.azure

Description

Azure contains settings specific to the Azure infrastructure provider.

Type

object

15.1.9. .spec.platformSpec.baremetal

Description

BareMetal contains settings specific to the BareMetal platform.

Type

object

Property	Type	Description
apiServerInternalIPs	**array (string)**	apiServerInternalIPs are the IP addresses to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. These are the IPs for a self-hosted load balancer in front of the API servers. In dual stack clusters this list contains two IP addresses, one from IPv4 family and one from IPv6. In single stack clusters a single IP address is expected. When omitted, values from the status.apiServerInternalIPs will be used. Once set, the list cannot be completely removed (but its second entry can).
ingressIPs	**array (string)**	ingressIPs are the external IPs which route to the default ingress controller. The IPs are suitable targets of a wildcard DNS record used to resolve default route host names. In dual stack clusters this list contains two IP addresses, one from IPv4 family and one from IPv6. In single stack clusters a single IP address is expected. When omitted, values from the status.ingressIPs will be used. Once set, the list cannot be completely removed (but its second entry can).

Property	Type	Description
machineNetworks	**array (string)**	machineNetworks are IP networks used to connect all the OpenShift cluster nodes. Each network is provided in the CIDR format and should be IPv4 or IPv6, for example "10.0.0.0/8" or "fd00::/8".

15.1.10. .spec.platformSpec.equinixMetal

Description

EquinixMetal contains settings specific to the Equinix Metal infrastructure provider.

Type

object

15.1.11. .spec.platformSpec.external

Description

ExternalPlatformType represents generic infrastructure provider. Platform-specific components should be supplemented separately.

Type

object

Property	Type	Description
platformName	**string**	PlatformName holds the arbitrary string representing the infrastructure provider name, expected to be set at the installation time. This field is solely for informational and reporting purposes and is not expected to be used for decision-making.

15.1.12. .spec.platformSpec.gcp

Description

GCP contains settings specific to the Google Cloud Platform infrastructure provider.

Type

object

15.1.13. .spec.platformSpec.ibmcloud

Description

IBMCloud contains settings specific to the IBMCloud infrastructure provider.

Type

object

15.1.14. .spec.platformSpec.kubevirt

Description

Kubevirt contains settings specific to the kubevirt infrastructure provider.

Type

object

15.1.15. .spec.platformSpec.nutanix

Description

Nutanix contains settings specific to the Nutanix infrastructure provider.

Type

object

Required

- **prismCentral**
- **prismElements**

Property	Type	Description
failureDomains	**array**	failureDomains configures failure domains information for the Nutanix platform. When set, the failure domains defined here may be used to spread Machines across prism element clusters to improve fault tolerance of the cluster.
failureDomains[]	**object**	NutanixFailureDomain configures failure domain information for the Nutanix platform.
prismCentral	**object**	prismCentral holds the endpoint address and port to access the Nutanix Prism Central. When a cluster-wide proxy is installed, by default, this endpoint will be accessed via the proxy. Should you wish for communication with this endpoint not to be proxied, please add the endpoint to the proxy spec.noProxy list.

Property	Type	Description
prismElements	**array**	prismElements holds one or more endpoint address and port data to access the Nutanix Prism Elements (clusters) of the Nutanix Prism Central. Currently we only support one Prism Element (cluster) for an OpenShift cluster, where all the Nutanix resources (VMs, subnets, volumes, etc.) used in the OpenShift cluster are located. In the future, we may support Nutanix resources (VMs, etc.) spread over multiple Prism Elements (clusters) of the Prism Central.
prismElements[]	**object**	NutanixPrismElementEndpoint holds the name and endpoint data for a Prism Element (cluster)

15.1.16. .spec.platformSpec.nutanix.failureDomains

Description

failureDomains configures failure domains information for the Nutanix platform. When set, the failure domains defined here may be used to spread Machines across prism element clusters to improve fault tolerance of the cluster.

Type

array

15.1.17. .spec.platformSpec.nutanix.failureDomains[]

Description

NutanixFailureDomain configures failure domain information for the Nutanix platform.

Type

object

Required

- **cluster**
- **name**
- **subnets**

Property	Type	Description
cluster	**object**	cluster is to identify the cluster (the Prism Element under management of the Prism Central), in which the Machine's VM will be created. The cluster identifier (uuid or name) can be obtained from the Prism Central console or using the prism_central API.
name	**string**	name defines the unique name of a failure domain. Name is required and must be at most 64 characters in length. It must consist of only lower case alphanumeric characters and hyphens (-). It must start and end with an alphanumeric character. This value is arbitrary and is used to identify the failure domain within the platform.
subnets	**array**	subnets holds a list of identifiers (one or more) of the cluster's network subnets If the feature gate NutanixMultiSubnets is enabled, up to 32 subnets may be configured. for the Machine's VM to connect to. The subnet identifiers (uuid or name) can be obtained from the Prism Central console or using the prism_central API.
subnets[]	**object**	NutanixResourceIdentifier holds the identity of a Nutanix PC resource (cluster, image, subnet, etc.)

15.1.18. .spec.platformSpec.nutanix.failureDomains[].cluster

Description

cluster is to identify the cluster (the Prism Element under management of the Prism Central), in which the Machine's VM will be created. The cluster identifier (uuid or name) can be obtained from the Prism Central console or using the prism_central API.

Type

object

Required

- **type**

- **type**

Property	Type	Description
name	**string**	name is the resource name in the PC. It cannot be empty if the type is Name.
type	**string**	type is the identifier type to use for this resource.
uuid	**string**	uuid is the UUID of the resource in the PC. It cannot be empty if the type is UUID.

15.1.19. .spec.platformSpec.nutanix.failureDomains[].subnets

Description

subnets holds a list of identifiers (one or more) of the cluster's network subnets If the feature gate NutanixMultiSubnets is enabled, up to 32 subnets may be configured. for the Machine's VM to connect to. The subnet identifiers (uuid or name) can be obtained from the Prism Central console or using the prism_central API.

Type

array

15.1.20. .spec.platformSpec.nutanix.failureDomains[].subnets[]

Description

NutanixResourceIdentifier holds the identity of a Nutanix PC resource (cluster, image, subnet, etc.)

Type

object

Required

- **type**

Property	Type	Description
name	**string**	name is the resource name in the PC. It cannot be empty if the type is Name.
type	**string**	type is the identifier type to use for this resource.
uuid	**string**	uuid is the UUID of the resource in the PC. It cannot be empty if the type is UUID.

15.1.21. .spec.platformSpec.nutanix.prismCentral

Description

prismCentral holds the endpoint address and port to access the Nutanix Prism Central. When a cluster-wide proxy is installed, by default, this endpoint will be accessed via the proxy. Should you wish for communication with this endpoint not to be proxied, please add the endpoint to the proxy spec.noProxy list.

Type

object

Required

- **address**
- **port**

Property	Type	Description
address	**string**	address is the endpoint address (DNS name or IP address) of the Nutanix Prism Central or Element (cluster)
port	**integer**	port is the port number to access the Nutanix Prism Central or Element (cluster)

15.1.22. .spec.platformSpec.nutanix.prismElements

Description

prismElements holds one or more endpoint address and port data to access the Nutanix Prism Elements (clusters) of the Nutanix Prism Central. Currently we only support one Prism Element (cluster) for an OpenShift cluster, where all the Nutanix resources (VMs, subnets, volumes, etc.) used in the OpenShift cluster are located. In the future, we may support Nutanix resources (VMs, etc.) spread over multiple Prism Elements (clusters) of the Prism Central.

Type

array

15.1.23. .spec.platformSpec.nutanix.prismElements[]

Description

NutanixPrismElementEndpoint holds the name and endpoint data for a Prism Element (cluster)

Type

object

Required

- **endpoint**
- **name**

Property	Type	Description
endpoint	**object**	endpoint holds the endpoint address and port data of the Prism Element (cluster). When a cluster-wide proxy is installed, by default, this endpoint will be accessed via the proxy. Should you wish for communication with this endpoint not to be proxied, please add the endpoint to the proxy spec.noProxy list.
name	**string**	name is the name of the Prism Element (cluster). This value will correspond with the cluster field configured on other resources (eg Machines, PVCs, etc).

15.1.24. .spec.platformSpec.nutanix.prismElements[].endpoint

Description

endpoint holds the endpoint address and port data of the Prism Element (cluster). When a cluster-wide proxy is installed, by default, this endpoint will be accessed via the proxy. Should you wish for communication with this endpoint not to be proxied, please add the endpoint to the proxy spec.noProxy list.

Type

object

Required

- **address**
- **port**

Property	Type	Description
address	**string**	address is the endpoint address (DNS name or IP address) of the Nutanix Prism Central or Element (cluster)
port	**integer**	port is the port number to access the Nutanix Prism Central or Element (cluster)

15.1.25. .spec.platformSpec.openstack

Description

OpenStack contains settings specific to the OpenStack infrastructure provider.

Type

object

Property	Type	Description
apiServerInternalIPs	**array (string)**	apiServerInternalIPs are the IP addresses to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. These are the IPs for a self-hosted load balancer in front of the API servers. In dual stack clusters this list contains two IP addresses, one from IPv4 family and one from IPv6. In single stack clusters a single IP address is expected. When omitted, values from the status.apiServerInternalIPs will be used. Once set, the list cannot be completely removed (but its second entry can).
ingressIPs	**array (string)**	ingressIPs are the external IPs which route to the default ingress controller. The IPs are suitable targets of a wildcard DNS record used to resolve default route host names. In dual stack clusters this list contains two IP addresses, one from IPv4 family and one from IPv6. In single stack clusters a single IP address is expected. When omitted, values from the status.ingressIPs will be used. Once set, the list cannot be completely removed (but its second entry can).
machineNetworks	**array (string)**	machineNetworks are IP networks used to connect all the OpenShift cluster nodes. Each network is provided in the CIDR format and should be IPv4 or IPv6, for example "10.0.0.0/8" or "fd00::/8".

15.1.26. .spec.platformSpec.ovirt

Description

Ovirt contains settings specific to the oVirt infrastructure provider.

Type

object

15.1.27. .spec.platformSpec.powervs

Description

PowerVS contains settings specific to the IBM Power Systems Virtual Servers infrastructure provider.

Type

object

Property	Type	Description
serviceEndpoints	**array**	serviceEndpoints is a list of custom endpoints which will override the default service endpoints of a Power VS service.
serviceEndpoints[]	**object**	PowervsServiceEndpoint stores the configuration of a custom url to override existing defaults of PowerVS Services.

15.1.28. .spec.platformSpec.powervs.serviceEndpoints

Description

serviceEndpoints is a list of custom endpoints which will override the default service endpoints of a Power VS service.

Type

array

15.1.29. .spec.platformSpec.powervs.serviceEndpoints[]

Description

PowervsServiceEndpoint stores the configuration of a custom url to override existing defaults of PowerVS Services.

Type

object

Required

- **name**
- **url**

Property	Type	Description
name	**string**	name is the name of the Power VS service. Few of the services are IAM - https://cloud.ibm.com/apidocs/iam-identity-token-api ResourceController - https://cloud.ibm.com/apidocs/resource-controller/resource-controller Power Cloud - https://cloud.ibm.com/apidocs/power-cloud
url	**string**	url is fully qualified URI with scheme https, that overrides the default generated endpoint for a client. This must be provided and cannot be empty.

15.1.30. .spec.platformSpec.vsphere

Description

VSphere contains settings specific to the VSphere infrastructure provider.

Type

object

Property	Type	Description
apiServerInternalIPs	**array (string)**	apiServerInternalIPs are the IP addresses to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. These are the IPs for a self-hosted load balancer in front of the API servers. In dual stack clusters this list contains two IP addresses, one from IPv4 family and one from IPv6. In single stack clusters a single IP address is expected. When omitted, values from the status.apiServerInternalIPs will be used. Once set, the list cannot be completely removed (but its second entry can).

Property	Type	Description
failureDomains	**array**	failureDomains contains the definition of region, zone and the vCenter topology. If this is omitted failure domains (regions and zones) will not be used.
failureDomains[]	**object**	VSpherePlatformFailureDomainSpec holds the region and zone failure domain and the vCenter topology of that failure domain.
ingressIPs	**array (string)**	ingressIPs are the external IPs which route to the default ingress controller. The IPs are suitable targets of a wildcard DNS record used to resolve default route host names. In dual stack clusters this list contains two IP addresses, one from IPv4 family and one from IPv6. In single stack clusters a single IP address is expected. When omitted, values from the status.ingressIPs will be used. Once set, the list cannot be completely removed (but its second entry can).
machineNetworks	**array (string)**	machineNetworks are IP networks used to connect all the OpenShift cluster nodes. Each network is provided in the CIDR format and should be IPv4 or IPv6, for example "10.0.0.0/8" or "fd00::/8".
nodeNetworking	**object**	nodeNetworking contains the definition of internal and external network constraints for assigning the node's networking. If this field is omitted, networking defaults to the legacy address selection behavior which is to only support a single address and return the first one found.

Property	Type	Description
vcenters	**array**	vcenters holds the connection details for services to communicate with vCenter. Currently, only a single vCenter is supported, but in tech preview 3 vCenters are supported. Once the cluster has been installed, you are unable to change the current number of defined vCenters except in the case where the cluster has been upgraded from a version of OpenShift where the vsphere platform spec was not present. You may make modifications to the existing vCenters that are defined in the vcenters list in order to match with any added or modified failure domains.
vcenters[]	**object**	VSpherePlatformVCenterSpec stores the vCenter connection fields. This is used by the vSphere CCM.

15.1.31. .spec.platformSpec.vsphere.failureDomains

Description

failureDomains contains the definition of region, zone and the vCenter topology. If this is omitted failure domains (regions and zones) will not be used.

Type

array

15.1.32. .spec.platformSpec.vsphere.failureDomains[]

Description

VSpherePlatformFailureDomainSpec holds the region and zone failure domain and the vCenter topology of that failure domain.

Type

object

Required

- **name**
- **region**
- **server**

- **topology**
- **zone**

Property	Type	Description
name	**string**	name defines the arbitrary but unique name of a failure domain.
region	**string**	region defines the name of a region tag that will be attached to a vCenter datacenter. The tag category in vCenter must be named openshift-region.
server	**string**	server is the fully-qualified domain name or the IP address of the vCenter server.
topology	**object**	Topology describes a given failure domain using vSphere constructs
zone	**string**	zone defines the name of a zone tag that will be attached to a vCenter cluster. The tag category in vCenter must be named openshift-zone.

15.1.33. .spec.platformSpec.vsphere.failureDomains[].topology

Description

Topology describes a given failure domain using vSphere constructs

Type

object

Required

- **computeCluster**
- **datacenter**
- **datastore**
- **networks**

Property	Type	Description

Property	Type	Description
computeCluster	**string**	computeCluster the absolute path of the vCenter cluster in which virtual machine will be located. The absolute path is of the form /<datacenter>/host/<cluster>. The maximum length of the path is 2048 characters.
datacenter	**string**	datacenter is the name of vCenter datacenter in which virtual machines will be located. The maximum length of the datacenter name is 80 characters.
datastore	**string**	datastore is the absolute path of the datastore in which the virtual machine is located. The absolute path is of the form /<datacenter>/datastore/<datastore> The maximum length of the path is 2048 characters.
folder	**string**	folder is the absolute path of the folder where virtual machines are located. The absolute path is of the form /<datacenter>/vm/<folder>. The maximum length of the path is 2048 characters.
networks	**array (string)**	networks is the list of port group network names within this failure domain. If feature gate VSphereMultiNetworks is enabled, up to 10 network adapters may be defined. 10 is the maximum number of virtual network devices which may be attached to a VM as defined by: https://configmax.esp.vmware.com/guest?vmwareproduct=vSphere&release=vSphere%208.0&categories=1-0 The available networks (port groups) can be listed using **govc ls 'network/*'** Networks should be in the form of an absolute path: /<datacenter>/network/<portgroup>.

Property	Type	Description
resourcePool	**string**	resourcePool is the absolute path of the resource pool where virtual machines will be created. The absolute path is of the form /<datacenter>/host/<cluster>/Resources/<resourcepool>. The maximum length of the path is 2048 characters.
template	**string**	template is the full inventory path of the virtual machine or template that will be cloned when creating new machines in this failure domain. The maximum length of the path is 2048 characters. When omitted, the template will be calculated by the control plane machineset operator based on the region and zone defined in VSpherePlatformFailureDomainSpec. For example, for zone=zonea, region=region1, and infrastructure name=test, the template path would be calculated as /<datacenter>/vm/test-rhcos-region1-zonea.

15.1.34. .spec.platformSpec.vsphere.nodeNetworking

Description

nodeNetworking contains the definition of internal and external network constraints for assigning the node's networking. If this field is omitted, networking defaults to the legacy address selection behavior which is to only support a single address and return the first one found.

Type

object

Property	Type	Description
external	**object**	external represents the network configuration of the node that is externally routable.
internal	**object**	internal represents the network configuration of the node that is routable only within the cluster.

15.1.35. .spec.platformSpec.vsphere.nodeNetworking.external

Description

external represents the network configuration of the node that is externally routable.

Type

object

Property	Type	Description
excludeNetworkSubnetCidr	**array (string)**	excludeNetworkSubnetCidr IP addresses in subnet ranges will be excluded when selecting the IP address from the VirtualMachine's VM for use in the status.addresses fields.
network	**string**	network VirtualMachine's VM Network names that will be used to when searching for status.addresses fields. Note that if internal.networkSubnetCIDR and external.networkSubnetCIDR are not set, then the vNIC associated to this network must only have a single IP address assigned to it. The available networks (port groups) can be listed using **govc ls 'network/*'**
networkSubnetCidr	**array (string)**	networkSubnetCidr IP address on VirtualMachine's network interfaces included in the fields' CIDRs that will be used in respective status.addresses fields.

15.1.36. .spec.platformSpec.vsphere.nodeNetworking.internal

Description

internal represents the network configuration of the node that is routable only within the cluster.

Type

object

Property	Type	Description

Property	Type	Description
excludeNetworkSubnetCidr	**array (string)**	excludeNetworkSubnetCidr IP addresses in subnet ranges will be excluded when selecting the IP address from the VirtualMachine's VM for use in the status.addresses fields.
network	**string**	network VirtualMachine's VM Network names that will be used to when searching for status.addresses fields. Note that if internal.networkSubnetCIDR and external.networkSubnetCIDR are not set, then the vNIC associated to this network must only have a single IP address assigned to it. The available networks (port groups) can be listed using **govc ls 'network/*'**
networkSubnetCidr	**array (string)**	networkSubnetCidr IP address on VirtualMachine's network interfaces included in the fields' CIDRs that will be used in respective status.addresses fields.

15.1.37. .spec.platformSpec.vsphere.vcenters

Description

vcenters holds the connection details for services to communicate with vCenter. Currently, only a single vCenter is supported, but in tech preview 3 vCenters are supported. Once the cluster has been installed, you are unable to change the current number of defined vCenters except in the case where the cluster has been upgraded from a version of OpenShift where the vsphere platform spec was not present. You may make modifications to the existing vCenters that are defined in the vcenters list in order to match with any added or modified failure domains.

Type

array

15.1.38. .spec.platformSpec.vsphere.vcenters[]

Description

VSpherePlatformVCenterSpec stores the vCenter connection fields. This is used by the vSphere CCM.

Type

object

Required

- **datacenters**
- **server**

Property	Type	Description
datacenters	**array (string)**	The vCenter Datacenters in which the RHCOS vm guests are located. This field will be used by the Cloud Controller Manager. Each datacenter listed here should be used within a topology.
port	**integer**	port is the TCP port that will be used to communicate to the vCenter endpoint. When omitted, this means the user has no opinion and it is up to the platform to choose a sensible default, which is subject to change over time.
server	**string**	server is the fully-qualified domain name or the IP address of the vCenter server.

15.1.39. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

Property	Type	Description
apiServerInternalURI	**string**	apiServerInternalURL is a valid URI with scheme 'https', address and optionally a port (defaulting to 443). apiServerInternalURL can be used by components like kubelets, to contact the Kubernetes API server using the infrastructure provider rather than Kubernetes networking.

Property	Type	Description
apiServerURL	**string**	apiServerURL is a valid URI with scheme 'https', address and optionally a port (defaulting to 443). apiServerURL can be used by components like the web console to tell users where to find the Kubernetes API.
controlPlaneTopology	**string**	controlPlaneTopology expresses the expectations for operands that normally run on control nodes. The default is 'HighlyAvailable', which represents the behavior operators have in a "normal" cluster. The 'SingleReplica' mode will be used in single-node deployments and the operators should not configure the operand for highly-available operation The 'External' mode indicates that the control plane is hosted externally to the cluster and that its components are not visible within the cluster.
cpuPartitioning	**string**	cpuPartitioning expresses if CPU partitioning is a currently enabled feature in the cluster. CPU Partitioning means that this cluster can support partitioning workloads to specific CPU Sets. Valid values are "None" and "AllNodes". When omitted, the default value is "None". The default value of "None" indicates that no nodes will be setup with CPU partitioning. The "AllNodes" value indicates that all nodes have been setup with CPU partitioning, and can then be further configured via the PerformanceProfile API.

Property	Type	Description
etcdDiscoveryDomain	**string**	etcdDiscoveryDomain is the domain used to fetch the SRV records for discovering etcd servers and clients. For more info: https://github.com/etcd-io/etcd/blob/329be66e8b3f9e2e6af83c123ff89297e49ebd15/Documentation/op-guide/clustering.md#dns-discovery deprecated: as of 4.7, this field is no longer set or honored. It will be removed in a future release.
infrastructureName	**string**	infrastructureName uniquely identifies a cluster with a human friendly name. Once set it should not be changed. Must be of max length 27 and must have only alphanumeric or hyphen characters.
infrastructureTopology	**string**	infrastructureTopology expresses the expectations for infrastructure services that do not run on control plane nodes, usually indicated by a node selector for a **role** value other than **master**. The default is 'HighlyAvailable', which represents the behavior operators have in a "normal" cluster. The 'SingleReplica' mode will be used in single-node deployments and the operators should not configure the operand for highly-available operation NOTE: External topology mode is not applicable for this field.
platform	**string**	platform is the underlying infrastructure provider for the cluster. Deprecated: Use platformStatus.type instead.

Property	Type	Description
platformStatus	**object**	platformStatus holds status information specific to the underlying infrastructure provider.

15.1.40. .status.platformStatus

Description

platformStatus holds status information specific to the underlying infrastructure provider.

Type

object

Property	Type	Description
alibabaCloud	**object**	AlibabaCloud contains settings specific to the Alibaba Cloud infrastructure provider.
aws	**object**	AWS contains settings specific to the Amazon Web Services infrastructure provider.
azure	**object**	Azure contains settings specific to the Azure infrastructure provider.
baremetal	**object**	BareMetal contains settings specific to the BareMetal platform.
equinixMetal	**object**	EquinixMetal contains settings specific to the Equinix Metal infrastructure provider.
external	**object**	External contains settings specific to the generic External infrastructure provider.
gcp	**object**	GCP contains settings specific to the Google Cloud Platform infrastructure provider.
ibmcloud	**object**	IBMCloud contains settings specific to the IBMCloud infrastructure provider.

Property	Type	Description
kubevirt	**object**	Kubevirt contains settings specific to the kubevirt infrastructure provider.
nutanix	**object**	Nutanix contains settings specific to the Nutanix infrastructure provider.
openstack	**object**	OpenStack contains settings specific to the OpenStack infrastructure provider.
ovirt	**object**	Ovirt contains settings specific to the oVirt infrastructure provider.
powervs	**object**	PowerVS contains settings specific to the Power Systems Virtual Servers infrastructure provider.
type	**string**	type is the underlying infrastructure provider for the cluster. This value controls whether infrastructure automation such as service load balancers, dynamic volume provisioning, machine creation and deletion, and other integrations are enabled. If None, no infrastructure automation is enabled. Allowed values are "AWS", "Azure", "BareMetal", "GCP", "Libvirt", "OpenStack", "VSphere", "oVirt", "EquinixMetal", "PowerVS", "AlibabaCloud", "Nutanix" and "None". Individual components may not support all platforms, and must handle unrecognized platforms as None if they do not support that platform. This value will be synced with to the **status.platform** and **status.platformStatus.type**. Currently this value cannot be changed once set.

Property	Type	Description
vsphere	**object**	VSphere contains settings specific to the VSphere infrastructure provider.

15.1.41. .status.platformStatus.alibabaCloud

Description

AlibabaCloud contains settings specific to the Alibaba Cloud infrastructure provider.

Type

object

Required

- **region**

Property	Type	Description
region	**string**	region specifies the region for Alibaba Cloud resources created for the cluster.
resourceGroupID	**string**	resourceGroupID is the ID of the resource group for the cluster.
resourceTags	**array**	resourceTags is a list of additional tags to apply to Alibaba Cloud resources created for the cluster.
resourceTags[]	**object**	AlibabaCloudResourceTag is the set of tags to add to apply to resources.

15.1.42. .status.platformStatus.alibabaCloud.resourceTags

Description

resourceTags is a list of additional tags to apply to Alibaba Cloud resources created for the cluster.

Type

array

15.1.43. .status.platformStatus.alibabaCloud.resourceTags[]

Description

AlibabaCloudResourceTag is the set of tags to add to apply to resources.

Type

object

Required

- **key**
- **value**

Property	Type	Description
key	**string**	key is the key of the tag.
value	**string**	value is the value of the tag.

15.1.44. .status.platformStatus.aws

Description

AWS contains settings specific to the Amazon Web Services infrastructure provider.

Type

object

Property	Type	Description
region	**string**	region holds the default AWS region for new AWS resources created by the cluster.
resourceTags	**array**	resourceTags is a list of additional tags to apply to AWS resources created for the cluster. See https://docs.aws.amazon.com/general/latest/gr/aws_tagging.html for information on tagging AWS resources. AWS supports a maximum of 50 tags per resource. OpenShift reserves 25 tags for its use, leaving 25 tags available for the user.
resourceTags[]	**object**	AWSResourceTag is a tag to apply to AWS resources created for the cluster.
serviceEndpoints	**array**	ServiceEndpoints list contains custom endpoints which will override default service endpoint of AWS Services. There must be only one ServiceEndpoint for a service.

Property	Type	Description
serviceEndpoints[]	**object**	AWSServiceEndpoint store the configuration of a custom url to override existing defaults of AWS Services.

15.1.45. .status.platformStatus.aws.resourceTags

Description

resourceTags is a list of additional tags to apply to AWS resources created for the cluster. See https://docs.aws.amazon.com/general/latest/gr/aws_tagging.html for information on tagging AWS resources. AWS supports a maximum of 50 tags per resource. OpenShift reserves 25 tags for its use, leaving 25 tags available for the user.

Type

array

15.1.46. .status.platformStatus.aws.resourceTags[]

Description

AWSResourceTag is a tag to apply to AWS resources created for the cluster.

Type

object

Required

- **key**
- **value**

Property	Type	Description
key	**string**	key is the key of the tag
value	**string**	value is the value of the tag. Some AWS service do not support empty values. Since tags are added to resources in many services, the length of the tag value must meet the requirements of all services.

15.1.47. .status.platformStatus.aws.serviceEndpoints

Description

ServiceEndpoints list contains custom endpoints which will override default service endpoint of AWS Services. There must be only one ServiceEndpoint for a service.

Type

array

15.1.48. .status.platformStatus.aws.serviceEndpoints[]

Description

AWSServiceEndpoint store the configuration of a custom url to override existing defaults of AWS Services.

Type

object

Property	Type	Description
name	**string**	name is the name of the AWS service. The list of all the service names can be found at https://docs.aws.amazon.com/general/latest/gr/aws-service-information.html This must be provided and cannot be empty.
url	**string**	url is fully qualified URI with scheme https, that overrides the default generated endpoint for a client. This must be provided and cannot be empty.

15.1.49. .status.platformStatus.azure

Description

Azure contains settings specific to the Azure infrastructure provider.

Type

object

Property	Type	Description
armEndpoint	**string**	armEndpoint specifies a URL to use for resource management in non-soverign clouds such as Azure Stack.

Property	Type	Description
cloudName	**string**	cloudName is the name of the Azure cloud environment which can be used to configure the Azure SDK with the appropriate Azure API endpoints. If empty, the value is equal to **AzurePublicCloud**.
networkResourceGroupName	**string**	networkResourceGroupName is the Resource Group for network resources like the Virtual Network and Subnets used by the cluster. If empty, the value is same as ResourceGroupName.
resourceGroupName	**string**	resourceGroupName is the Resource Group for new Azure resources created for the cluster.
resourceTags	**array**	resourceTags is a list of additional tags to apply to Azure resources created for the cluster. See https://docs.microsoft.com/en-us/rest/api/resources/tags for information on tagging Azure resources. Due to limitations on Automation, Content Delivery Network, DNS Azure resources, a maximum of 15 tags may be applied. OpenShift reserves 5 tags for internal use, allowing 10 tags for user configuration.
resourceTags[]	**object**	AzureResourceTag is a tag to apply to Azure resources created for the cluster.

15.1.50. .status.platformStatus.azure.resourceTags

Description

resourceTags is a list of additional tags to apply to Azure resources created for the cluster. See https://docs.microsoft.com/en-us/rest/api/resources/tags for information on tagging Azure resources. Due to limitations on Automation, Content Delivery Network, DNS Azure resources, a maximum of 15 tags may be applied. OpenShift reserves 5 tags for internal use, allowing 10 tags for user configuration.

Type

array

15.1.51. .status.platformStatus.azure.resourceTags[]

Description

AzureResourceTag is a tag to apply to Azure resources created for the cluster.

Type

object

Required

- **key**
- **value**

Property	Type	Description
key	**string**	key is the key part of the tag. A tag key can have a maximum of 128 characters and cannot be empty. Key must begin with a letter, end with a letter, number or underscore, and must contain only alphanumeric characters and the following special characters **_ . -**.
value	**string**	value is the value part of the tag. A tag value can have a maximum of 256 characters and cannot be empty. Value must contain only alphanumeric characters and the following special characters **_ + , - . / : ; < = > ? @**.

15.1.52. .status.platformStatus.baremetal

Description

BareMetal contains settings specific to the BareMetal platform.

Type

object

Property	Type	Description

Property	Type	Description
apiServerInternalIP	**string**	apiServerInternalIP is an IP address to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. It is the IP that the Infrastructure.status.apiServerInternalURI points to. It is the IP for a self-hosted load balancer in front of the API servers. Deprecated: Use APIServerInternalIPs instead.
apiServerInternalIPs	**array (string)**	apiServerInternalIPs are the IP addresses to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. These are the IPs for a self-hosted load balancer in front of the API servers. In dual stack clusters this list contains two IPs otherwise only one.
ingressIP	**string**	ingressIP is an external IP which routes to the default ingress controller. The IP is a suitable target of a wildcard DNS record used to resolve default route host names. Deprecated: Use IngressIPs instead.
ingressIPs	**array (string)**	ingressIPs are the external IPs which route to the default ingress controller. The IPs are suitable targets of a wildcard DNS record used to resolve default route host names. In dual stack clusters this list contains two IPs otherwise only one.
loadBalancer	**object**	loadBalancer defines how the load balancer used by the cluster is configured.

Property	Type	Description
machineNetworks	**array (string)**	machineNetworks are IP networks used to connect all the OpenShift cluster nodes.
nodeDNSIP	**string**	nodeDNSIP is the IP address for the internal DNS used by the nodes. Unlike the one managed by the DNS operator, **NodeDNSIP** provides name resolution for the nodes themselves. There is no DNS-as-a-service for BareMetal deployments. In order to minimize necessary changes to the datacenter DNS, a DNS service is hosted as a static pod to serve those hostnames to the nodes in the cluster.

15.1.53. .status.platformStatus.baremetal.loadBalancer

Description

loadBalancer defines how the load balancer used by the cluster is configured.

Type

object

Property	Type	Description

Property	Type	Description
type	**string**	type defines the type of load balancer used by the cluster on BareMetal platform which can be a user-managed or openshift-managed load balancer that is to be used for the OpenShift API and Ingress endpoints. When set to OpenShiftManagedDefault the static pods in charge of API and Ingress traffic load-balancing defined in the machine config operator will be deployed. When set to UserManaged these static pods will not be deployed and it is expected that the load balancer is configured out of band by the deployer. When omitted, this means no opinion and the platform is left to choose a reasonable default. The default value is OpenShiftManagedDefault.

15.1.54. .status.platformStatus.equinixMetal

Description

EquinixMetal contains settings specific to the Equinix Metal infrastructure provider.

Type

object

Property	Type	Description
apiServerInternalIP	**string**	apiServerInternalIP is an IP address to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. It is the IP that the Infrastructure.status.apiServerInternalURI points to. It is the IP for a self-hosted load balancer in front of the API servers.

Property	Type	Description
ingressIP	**string**	ingressIP is an external IP which routes to the default ingress controller. The IP is a suitable target of a wildcard DNS record used to resolve default route host names.

15.1.55. .status.platformStatus.external

Description

External contains settings specific to the generic External infrastructure provider.

Type

object

Property	Type	Description
cloudControllerManager	**object**	cloudControllerManager contains settings specific to the external Cloud Controller Manager (a.k.a. CCM or CPI). When omitted, new nodes will be not tainted and no extra initialization from the cloud controller manager is expected.

15.1.56. .status.platformStatus.external.cloudControllerManager

Description

cloudControllerManager contains settings specific to the external Cloud Controller Manager (a.k.a. CCM or CPI). When omitted, new nodes will be not tainted and no extra initialization from the cloud controller manager is expected.

Type

object

Property	Type	Description

Property	Type	Description
state	**string**	state determines whether or not an external Cloud Controller Manager is expected to be installed within the cluster. https://kubernetes.io/docs/tasks/administer-cluster/running-cloud-controller/#running-cloud-controller-manager Valid values are "External", "None" and omitted. When set to "External", new nodes will be tainted as uninitialized when created, preventing them from running workloads until they are initialized by the cloud controller manager. When omitted or set to "None", new nodes will be not tainted and no extra initialization from the cloud controller manager is expected.

15.1.57. .status.platformStatus.gcp

Description

GCP contains settings specific to the Google Cloud Platform infrastructure provider.

Type

object

Property	Type	Description
projectID	**string**	resourceGroupName is the Project ID for new GCP resources created for the cluster.
region	**string**	region holds the region for new GCP resources created for the cluster.

Property	Type	Description
resourceLabels	**array**	resourceLabels is a list of additional labels to apply to GCP resources created for the cluster. See https://cloud.google.com/compute/docs/labeling-resources for information on labeling GCP resources. GCP supports a maximum of 64 labels per resource. OpenShift reserves 32 labels for internal use, allowing 32 labels for user configuration.
resourceLabels[]	**object**	GCPResourceLabel is a label to apply to GCP resources created for the cluster.
resourceTags	**array**	resourceTags is a list of additional tags to apply to GCP resources created for the cluster. See https://cloud.google.com/resource-manager/docs/tags/tags-overview for information on tagging GCP resources. GCP supports a maximum of 50 tags per resource.
resourceTags[]	**object**	GCPResourceTag is a tag to apply to GCP resources created for the cluster.

15.1.58. .status.platformStatus.gcp.resourceLabels

Description

resourceLabels is a list of additional labels to apply to GCP resources created for the cluster. See https://cloud.google.com/compute/docs/labeling-resources for information on labeling GCP resources. GCP supports a maximum of 64 labels per resource. OpenShift reserves 32 labels for internal use, allowing 32 labels for user configuration.

Type

array

15.1.59. .status.platformStatus.gcp.resourceLabels[]

Description

GCPResourceLabel is a label to apply to GCP resources created for the cluster.

Type

object

Required

- **key**
- **value**

Property	Type	Description
key	**string**	key is the key part of the label. A label key can have a maximum of 63 characters and cannot be empty. Label key must begin with a lowercase letter, and must contain only lowercase letters, numeric characters, and the following special characters _-. Label key must not have the reserved prefixes **kubernetes-io** and **openshift-io**.
value	**string**	value is the value part of the label. A label value can have a maximum of 63 characters and cannot be empty. Value must contain only lowercase letters, numeric characters, and the following special characters _-.

15.1.60. .status.platformStatus.gcp.resourceTags

Description

resourceTags is a list of additional tags to apply to GCP resources created for the cluster. See https://cloud.google.com/resource-manager/docs/tags/tags-overview for information on tagging GCP resources. GCP supports a maximum of 50 tags per resource.

Type

array

15.1.61. .status.platformStatus.gcp.resourceTags[]

Description

GCPResourceTag is a tag to apply to GCP resources created for the cluster.

Type

object

Required

- **key**
- **parentID**
- **value**

Property	Type	Description
key	**string**	key is the key part of the tag. A tag key can have a maximum of 63 characters and cannot be empty. Tag key must begin and end with an alphanumeric character, and must contain only uppercase, lowercase alphanumeric characters, and the following special characters **._-**.
parentID	**string**	parentID is the ID of the hierarchical resource where the tags are defined, e.g. at the Organization or the Project level. To find the Organization or Project ID refer to the following pages: https://cloud.google.com/resource-manager/docs/creating-managing-organization#retrieving_your_organization_id, https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects. An OrganizationID must consist of decimal numbers, and cannot have leading zeroes. A ProjectID must be 6 to 30 characters in length, can only contain lowercase letters, numbers, and hyphens, and must start with a letter, and cannot end with a hyphen.
value	**string**	value is the value part of the tag. A tag value can have a maximum of 63 characters and cannot be empty. Tag value must begin and end with an alphanumeric character, and must contain only uppercase, lowercase alphanumeric characters, and the following special characters **_-.@%=+:,*#&(){}[]** and spaces.

15.1.62. .status.platformStatus.ibmcloud

Description

IBMCloud contains settings specific to the IBMCloud infrastructure provider.

Type

object

Property	Type	Description
cisInstanceCRN	**string**	CISInstanceCRN is the CRN of the Cloud Internet Services instance managing the DNS zone for the cluster's base domain
dnsInstanceCRN	**string**	DNSInstanceCRN is the CRN of the DNS Services instance managing the DNS zone for the cluster's base domain
location	**string**	Location is where the cluster has been deployed
providerType	**string**	ProviderType indicates the type of cluster that was created
resourceGroupName	**string**	ResourceGroupName is the Resource Group for new IBMCloud resources created for the cluster.
serviceEndpoints	**array**	serviceEndpoints is a list of custom endpoints which will override the default service endpoints of an IBM Cloud service. These endpoints are consumed by components within the cluster to reach the respective IBM Cloud Services.
serviceEndpoints[]	**object**	IBMCloudServiceEndpoint stores the configuration of a custom url to override existing defaults of IBM Cloud Services.

15.1.63. .status.platformStatus.ibmcloud.serviceEndpoints

Description

serviceEndpoints is a list of custom endpoints which will override the default service endpoints of an IBM Cloud service. These endpoints are consumed by components within the cluster to reach the respective IBM Cloud Services.

Type

array

15.1.64. .status.platformStatus.ibmcloud.serviceEndpoints[]

Description

IBMCloudServiceEndpoint stores the configuration of a custom url to override existing defaults of IBM Cloud Services.

Type

object

Required

- **name**
- **url**

Property	Type	Description
name	**string**	name is the name of the IBM Cloud service. Possible values are: CIS, COS, COSConfig, DNSServices, GlobalCatalog, GlobalSearch, GlobalTagging, HyperProtect, IAM, KeyProtect, ResourceController, ResourceManager, or VPC. For example, the IBM Cloud Private IAM service could be configured with the service **name** of **IAM** and **url** of **https://private.iam.cloud.ibm.com** Whereas the IBM Cloud Private VPC service for US South (Dallas) could be configured with the service **name** of **VPC** and **url** of **https://us.south.private.iaas.cloud.ibm.com**
url	**string**	url is fully qualified URI with scheme https, that overrides the default generated endpoint for a client. This must be provided and cannot be empty.

15.1.65. .status.platformStatus.kubevirt

Description

Kubevirt contains settings specific to the kubevirt infrastructure provider.

Type

object

Property	Type	Description
apiServerInternalIP	**string**	apiServerInternalIP is an IP address to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. It is the IP that the Infrastructure.status.apiServerInternalURI points to. It is the IP for a self-hosted load balancer in front of the API servers.
ingressIP	**string**	ingressIP is an external IP which routes to the default ingress controller. The IP is a suitable target of a wildcard DNS record used to resolve default route host names.

15.1.66. .status.platformStatus.nutanix

Description

Nutanix contains settings specific to the Nutanix infrastructure provider.

Type

object

Property	Type	Description
apiServerInternalIP	**string**	apiServerInternalIP is an IP address to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. It is the IP that the Infrastructure.status.apiServerInternalURI points to. It is the IP for a self-hosted load balancer in front of the API servers. Deprecated: Use APIServerInternalIPs instead.

Property	Type	Description
apiServerInternalIPs	**array (string)**	apiServerInternalIPs are the IP addresses to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. These are the IPs for a self-hosted load balancer in front of the API servers. In dual stack clusters this list contains two IPs otherwise only one.
ingressIP	**string**	ingressIP is an external IP which routes to the default ingress controller. The IP is a suitable target of a wildcard DNS record used to resolve default route host names. Deprecated: Use IngressIPs instead.
ingressIPs	**array (string)**	ingressIPs are the external IPs which route to the default ingress controller. The IPs are suitable targets of a wildcard DNS record used to resolve default route host names. In dual stack clusters this list contains two IPs otherwise only one.
loadBalancer	**object**	loadBalancer defines how the load balancer used by the cluster is configured.

15.1.67. .status.platformStatus.nutanix.loadBalancer

Description

loadBalancer defines how the load balancer used by the cluster is configured.

Type

object

Property	Type	Description

Property	Type	Description
type	**string**	type defines the type of load balancer used by the cluster on Nutanix platform which can be a user-managed or openshift-managed load balancer that is to be used for the OpenShift API and Ingress endpoints. When set to OpenShiftManagedDefault the static pods in charge of API and Ingress traffic load-balancing defined in the machine config operator will be deployed. When set to UserManaged these static pods will not be deployed and it is expected that the load balancer is configured out of band by the deployer. When omitted, this means no opinion and the platform is left to choose a reasonable default. The default value is OpenShiftManagedDefault.

15.1.68. .status.platformStatus.openstack

Description

OpenStack contains settings specific to the OpenStack infrastructure provider.

Type

object

Property	Type	Description
apiServerInternalIP	**string**	apiServerInternalIP is an IP address to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. It is the IP that the Infrastructure.status.apiServerInternalURI points to. It is the IP for a self-hosted load balancer in front of the API servers. Deprecated: Use APIServerInternalIPs instead.

Property	Type	Description
apiServerInternalIPs	**array (string)**	apiServerInternalIPs are the IP addresses to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. These are the IPs for a self-hosted load balancer in front of the API servers. In dual stack clusters this list contains two IPs otherwise only one.
cloudName	**string**	cloudName is the name of the desired OpenStack cloud in the client configuration file (**clouds.yaml**).
ingressIP	**string**	ingressIP is an external IP which routes to the default ingress controller. The IP is a suitable target of a wildcard DNS record used to resolve default route host names. Deprecated: Use IngressIPs instead.
ingressIPs	**array (string)**	ingressIPs are the external IPs which route to the default ingress controller. The IPs are suitable targets of a wildcard DNS record used to resolve default route host names. In dual stack clusters this list contains two IPs otherwise only one.
loadBalancer	**object**	loadBalancer defines how the load balancer used by the cluster is configured.
machineNetworks	**array (string)**	machineNetworks are IP networks used to connect all the OpenShift cluster nodes.

Property	Type	Description
nodeDNSIP	**string**	nodeDNSIP is the IP address for the internal DNS used by the nodes. Unlike the one managed by the DNS operator, **NodeDNSIP** provides name resolution for the nodes themselves. There is no DNS-as-a-service for OpenStack deployments. In order to minimize necessary changes to the datacenter DNS, a DNS service is hosted as a static pod to serve those hostnames to the nodes in the cluster.

15.1.69. .status.platformStatus.openstack.loadBalancer

Description

loadBalancer defines how the load balancer used by the cluster is configured.

Type

object

Property	Type	Description
type	**string**	type defines the type of load balancer used by the cluster on OpenStack platform which can be a user-managed or openshift-managed load balancer that is to be used for the OpenShift API and Ingress endpoints. When set to OpenShiftManagedDefault the static pods in charge of API and Ingress traffic load-balancing defined in the machine config operator will be deployed. When set to UserManaged these static pods will not be deployed and it is expected that the load balancer is configured out of band by the deployer. When omitted, this means no opinion and the platform is left to choose a reasonable default. The default value is OpenShiftManagedDefault.

15.1.70. .status.platformStatus.ovirt

Description

Ovirt contains settings specific to the oVirt infrastructure provider.

Type

object

Property	Type	Description
apiServerInternalIP	**string**	apiServerInternalIP is an IP address to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. It is the IP that the Infrastructure.status.apiServerInternalURI points to. It is the IP for a self-hosted load balancer in front of the API servers. Deprecated: Use APIServerInternalIPs instead.
apiServerInternalIPs	**array (string)**	apiServerInternalIPs are the IP addresses to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. These are the IPs for a self-hosted load balancer in front of the API servers. In dual stack clusters this list contains two IPs otherwise only one.
ingressIP	**string**	ingressIP is an external IP which routes to the default ingress controller. The IP is a suitable target of a wildcard DNS record used to resolve default route host names. Deprecated: Use IngressIPs instead.

Property	Type	Description
ingressIPs	**array (string)**	ingressIPs are the external IPs which route to the default ingress controller. The IPs are suitable targets of a wildcard DNS record used to resolve default route host names. In dual stack clusters this list contains two IPs otherwise only one.
loadBalancer	**object**	loadBalancer defines how the load balancer used by the cluster is configured.
nodeDNSIP	**string**	deprecated: as of 4.6, this field is no longer set or honored. It will be removed in a future release.

15.1.71. .status.platformStatus.ovirt.loadBalancer

Description

loadBalancer defines how the load balancer used by the cluster is configured.

Type

object

Property	Type	Description

Property	Type	Description
type	**string**	type defines the type of load balancer used by the cluster on Ovirt platform which can be a user-managed or openshift-managed load balancer that is to be used for the OpenShift API and Ingress endpoints. When set to OpenShiftManagedDefault the static pods in charge of API and Ingress traffic load-balancing defined in the machine config operator will be deployed. When set to UserManaged these static pods will not be deployed and it is expected that the load balancer is configured out of band by the deployer. When omitted, this means no opinion and the platform is left to choose a reasonable default. The default value is OpenShiftManagedDefault.

15.1.72. .status.platformStatus.powervs

Description

PowerVS contains settings specific to the Power Systems Virtual Servers infrastructure provider.

Type

object

Property	Type	Description
cisInstanceCRN	**string**	CISInstanceCRN is the CRN of the Cloud Internet Services instance managing the DNS zone for the cluster's base domain
dnsInstanceCRN	**string**	DNSInstanceCRN is the CRN of the DNS Services instance managing the DNS zone for the cluster's base domain
region	**string**	region holds the default Power VS region for new Power VS resources created by the cluster.

Property	Type	Description
resourceGroup	**string**	resourceGroup is the resource group name for new IBMCloud resources created for a cluster. The resource group specified here will be used by cluster-image-registry-operator to set up a COS Instance in IBMCloud for the cluster registry. More about resource groups can be found here: https://cloud.ibm.com/docs/account?topic=account-rgs. When omitted, the image registry operator won't be able to configure storage, which results in the image registry cluster operator not being in an available state.
serviceEndpoints	**array**	serviceEndpoints is a list of custom endpoints which will override the default service endpoints of a Power VS service.
serviceEndpoints[]	**object**	PowervsServiceEndpoint stores the configuration of a custom url to override existing defaults of PowerVS Services.
zone	**string**	zone holds the default zone for the new Power VS resources created by the cluster. Note: Currently only single-zone OCP clusters are supported

15.1.73. .status.platformStatus.powervs.serviceEndpoints

Description

serviceEndpoints is a list of custom endpoints which will override the default service endpoints of a Power VS service.

Type

array

15.1.74. .status.platformStatus.powervs.serviceEndpoints[]

Description

PowervsServiceEndpoint stores the configuration of a custom url to override existing defaults of PowerVS Services.

Type

object

Required

- **name**
- **url**

Property	Type	Description
name	**string**	name is the name of the Power VS service. Few of the services are IAM - https://cloud.ibm.com/apidocs/iam-identity-token-api ResourceController - https://cloud.ibm.com/apidocs/resource-controller/resource-controller Power Cloud - https://cloud.ibm.com/apidocs/power-cloud
url	**string**	url is fully qualified URI with scheme https, that overrides the default generated endpoint for a client. This must be provided and cannot be empty.

15.1.75. .status.platformStatus.vsphere

Description

VSphere contains settings specific to the VSphere infrastructure provider.

Type

object

Property	Type	Description

Property	Type	Description
apiServerInternalIP	**string**	apiServerInternalIP is an IP address to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. It is the IP that the Infrastructure.status.apiServerInternalURI points to. It is the IP for a self-hosted load balancer in front of the API servers. Deprecated: Use APIServerInternalIPs instead.
apiServerInternalIPs	**array (string)**	apiServerInternalIPs are the IP addresses to contact the Kubernetes API server that can be used by components inside the cluster, like kubelets using the infrastructure rather than Kubernetes networking. These are the IPs for a self-hosted load balancer in front of the API servers. In dual stack clusters this list contains two IPs otherwise only one.
ingressIP	**string**	ingressIP is an external IP which routes to the default ingress controller. The IP is a suitable target of a wildcard DNS record used to resolve default route host names. Deprecated: Use IngressIPs instead.
ingressIPs	**array (string)**	ingressIPs are the external IPs which route to the default ingress controller. The IPs are suitable targets of a wildcard DNS record used to resolve default route host names. In dual stack clusters this list contains two IPs otherwise only one.
loadBalancer	**object**	loadBalancer defines how the load balancer used by the cluster is configured.

Property	Type	Description
machineNetworks	**array (string)**	machineNetworks are IP networks used to connect all the OpenShift cluster nodes.
nodeDNSIP	**string**	nodeDNSIP is the IP address for the internal DNS used by the nodes. Unlike the one managed by the DNS operator, **NodeDNSIP** provides name resolution for the nodes themselves. There is no DNS-as-a-service for vSphere deployments. In order to minimize necessary changes to the datacenter DNS, a DNS service is hosted as a static pod to serve those hostnames to the nodes in the cluster.

15.1.76. .status.platformStatus.vsphere.loadBalancer

Description

loadBalancer defines how the load balancer used by the cluster is configured.

Type

object

Property	Type	Description

Property	Type	Description
type	**string**	type defines the type of load balancer used by the cluster on VSphere platform which can be a user-managed or openshift-managed load balancer that is to be used for the OpenShift API and Ingress endpoints. When set to OpenShiftManagedDefault the static pods in charge of API and Ingress traffic load-balancing defined in the machine config operator will be deployed. When set to UserManaged these static pods will not be deployed and it is expected that the load balancer is configured out of band by the deployer. When omitted, this means no opinion and the platform is left to choose a reasonable default. The default value is OpenShiftManagedDefault.

15.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/infrastructures**
  - **DELETE**: delete collection of Infrastructure
  - **GET**: list objects of kind Infrastructure
  - **POST**: create an Infrastructure
- **/apis/config.openshift.io/v1/infrastructures/{name}**
  - **DELETE**: delete an Infrastructure
  - **GET**: read the specified Infrastructure
  - **PATCH**: partially update the specified Infrastructure
  - **PUT**: replace the specified Infrastructure
- **/apis/config.openshift.io/v1/infrastructures/{name}/status**
  - **GET**: read status of the specified Infrastructure
  - **PATCH**: partially update status of the specified Infrastructure
  - **PUT**: replace status of the specified Infrastructure

15.2.1. /apis/config.openshift.io/v1/infrastructures

HTTP method

DELETE

Description

delete collection of Infrastructure

Table 15.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Infrastructure

Table 15.2. HTTP responses

HTTP code	Reponse body
200 - OK	**InfrastructureList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an Infrastructure

Table 15.3. Query parameters

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

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

Table 15.4. Body parameters

Parameter	Type	Description
body	**Infrastructure** schema	

Table 15.5. HTTP responses

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

15.2.2. /apis/config.openshift.io/v1/infrastructures/{name}

Table 15.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Infrastructure

HTTP method

DELETE

Description

delete an Infrastructure

Table 15.7. 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 15.8. HTTP responses

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

HTTP method

GET

Description

read the specified Infrastructure

Table 15.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Infrastructure** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Infrastructure

Table 15.10. 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 15.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Infrastructure** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Infrastructure

Table 15.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 15.13. Body parameters

Parameter	Type	Description
body	**Infrastructure** schema	

Table 15.14. HTTP responses

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

15.2.3. /apis/config.openshift.io/v1/infrastructures/{name}/status

Table 15.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Infrastructure

HTTP method

GET

Description

read status of the specified Infrastructure

Table 15.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Infrastructure** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Infrastructure

Table 15.17. Query parameters

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

Table 15.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Infrastructure** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Infrastructure

Table 15.19. Query parameters

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

Table 15.20. Body parameters

Parameter	Type	Description
body	**Infrastructure** schema	

Table 15.21. HTTP responses

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

CHAPTER 16. INGRESS [CONFIG.OPENSHIFT.IO/V1]

Description

Ingress holds cluster-wide information about ingress, including the default ingress domain used for routes. The canonical name is **cluster**.

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

Type

object

Required

- **spec**

16.1. 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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

16.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
appsDomain	**string**	appsDomain is an optional domain to use instead of the one specified in the domain field when a Route is created without specifying an explicit host. If appsDomain is nonempty, this value is used to generate default host values for Route. Unlike domain, appsDomain may be modified after installation. This assumes a new ingresscontroller has been setup with a wildcard certificate.
componentRoutes	**array**	componentRoutes is an optional list of routes that are managed by OpenShift components that a cluster-admin is able to configure the hostname and serving certificate for. The namespace and name of each route in this list should match an existing entry in the status.componentRoutes list. To determine the set of configurable Routes, look at namespace and name of entries in the .status.componentRoutes list, where participating operators write the status of configurable routes.
componentRoutes[]	**object**	ComponentRouteSpec allows for configuration of a route's hostname and serving certificate.

Property	Type	Description
domain	**string**	domain is used to generate a default host name for a route when the route's host name is empty. The generated host name will follow this pattern: "<route-name>.<route-namespace>.<domain>". It is also used as the default wildcard domain suffix for ingress. The default ingresscontroller domain will follow this pattern: "*.<domain>". Once set, changing domain is not currently supported.
loadBalancer	**object**	loadBalancer contains the load balancer details in general which are not only specific to the underlying infrastructure provider of the current cluster and are required for Ingress Controller to work on OpenShift.
requiredHSTSPolicies	**array**	requiredHSTSPolicies specifies HSTS policies that are required to be set on newly created or updated routes matching the domainPattern/s and namespaceSelector/s that are specified in the policy. Each requiredHSTSPolicy must have at least a domainPattern and a maxAge to validate a route HSTS Policy route annotation, and affect route admission. A candidate route is checked for HSTS Policies if it has the HSTS Policy route annotation: "haproxy.router.openshift.io/hsts_header" E.g. haproxy.router.openshift.io/hsts_header: max-age=31536000;preload;includeSubDomains - For each candidate route, if it matches a requiredHSTSPolicy domainPattern and optional namespaceSelector, then the maxAge, preloadPolicy, and

Property	Type	Description
		includeSubdomainsPolicy must be valid to be admitted. Otherwise, the route is rejected. - The first match, by domainPattern and optional namespaceSelector, in the ordering of the RequiredHSTSPolicies determines the route's admission status. - If the candidate route doesn't match any requiredHSTSPolicy domainPattern and optional namespaceSelector, then it may use any HSTS Policy annotation. The HSTS policy configuration may be changed after routes have already been created. An update to a previously admitted route may then fail if the updated route does not conform to the updated HSTS policy configuration. However, changing the HSTS policy configuration will not cause a route that is already admitted to stop working. Note that if there are no RequiredHSTSPolicies, any HSTS Policy annotation on the route is valid.
requiredHSTSPolicies[]	**object**	

16.1.2. .spec.componentRoutes

Description

componentRoutes is an optional list of routes that are managed by OpenShift components that a cluster-admin is able to configure the hostname and serving certificate for. The namespace and name of each route in this list should match an existing entry in the status.componentRoutes list. To determine the set of configurable Routes, look at namespace and name of entries in the .status.componentRoutes list, where participating operators write the status of configurable routes.

Type

array

16.1.3. .spec.componentRoutes[]

Description

ComponentRouteSpec allows for configuration of a route's hostname and serving certificate.

Type

object

Required

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

Property	Type	Description
hostname	**string**	hostname is the hostname that should be used by the route.
name	**string**	name is the logical name of the route to customize. The namespace and name of this componentRoute must match a corresponding entry in the list of status.componentRoutes if the route is to be customized.
namespace	**string**	namespace is the namespace of the route to customize. The namespace and name of this componentRoute must match a corresponding entry in the list of status.componentRoutes if the route is to be customized.
servingCertKeyPairSecret	**object**	servingCertKeyPairSecret is a reference to a secret of type **kubernetes.io/tls** in the openshift-config namespace. The serving cert/key pair must match and will be used by the operator to fulfill the intent of serving with this name. If the custom hostname uses the default routing suffix of the cluster, the Secret specification for a serving certificate will not be needed.

16.1.4. .spec.componentRoutes[].servingCertKeyPairSecret

Description

servingCertKeyPairSecret is a reference to a secret of type **kubernetes.io/tls** in the openshift-config namespace. The serving cert/key pair must match and will be used by the operator to fulfill the intent of serving with this name. If the custom hostname uses the default routing suffix of the cluster, the Secret specification for a serving certificate will not be needed.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

16.1.5. .spec.loadBalancer

Description

loadBalancer contains the load balancer details in general which are not only specific to the underlying infrastructure provider of the current cluster and are required for Ingress Controller to work on OpenShift.

Type

object

Property	Type	Description
platform	**object**	platform holds configuration specific to the underlying infrastructure provider for the ingress load balancers. When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time.

16.1.6. .spec.loadBalancer.platform

Description

platform holds configuration specific to the underlying infrastructure provider for the ingress load balancers. When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time.

Type

object

Property	Type	Description
aws	**object**	aws contains settings specific to the Amazon Web Services infrastructure provider.

Property	Type	Description
type	**string**	type is the underlying infrastructure provider for the cluster. Allowed values are "AWS", "Azure", "BareMetal", "GCP", "Libvirt", "OpenStack", "VSphere", "oVirt", "KubeVirt", "EquinixMetal", "PowerVS", "AlibabaCloud", "Nutanix" and "None". Individual components may not support all platforms, and must handle unrecognized platforms as None if they do not support that platform.

16.1.7. .spec.loadBalancer.platform.aws

Description

aws contains settings specific to the Amazon Web Services infrastructure provider.

Type

object

Required

- **type**

Property	Type	Description

Property	Type	Description
type	**string**	type allows user to set a load balancer type. When this field is set the default ingresscontroller will get created using the specified LBType. If this field is not set then the default ingress controller of LBType Classic will be created. Valid values are: * "Classic": A Classic Load Balancer that makes routing decisions at either the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS). See the following for additional details: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/load-balancer-types.html#clb * "NLB": A Network Load Balancer that makes routing decisions at the transport layer (TCP/SSL). See the following for additional details: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/load-balancer-types.html#nlb

16.1.8. .spec.requiredHSTSPolicies

Description

requiredHSTSPolicies specifies HSTS policies that are required to be set on newly created or updated routes matching the domainPattern/s and namespaceSelector/s that are specified in the policy. Each requiredHSTSPolicy must have at least a domainPattern and a maxAge to validate a route HSTS Policy route annotation, and affect route admission.
A candidate route is checked for HSTS Policies if it has the HSTS Policy route annotation: "haproxy.router.openshift.io/hsts_header" E.g. haproxy.router.openshift.io/hsts_header: max-age=31536000;preload;includeSubDomains

- For each candidate route, if it matches a requiredHSTSPolicy domainPattern and optional namespaceSelector, then the maxAge, preloadPolicy, and includeSubdomainsPolicy must be valid to be admitted. Otherwise, the route is rejected.
- The first match, by domainPattern and optional namespaceSelector, in the ordering of the RequiredHSTSPolicies determines the route's admission status.

- If the candidate route doesn't match any requiredHSTSPolicy domainPattern and optional namespaceSelector, then it may use any HSTS Policy annotation.

The HSTS policy configuration may be changed after routes have already been created. An update to a previously admitted route may then fail if the updated route does not conform to the updated HSTS policy configuration. However, changing the HSTS policy configuration will not cause a route that is already admitted to stop working.

Note that if there are no RequiredHSTSPolicies, any HSTS Policy annotation on the route is valid.

Type

array

16.1.9. .spec.requiredHSTSPolicies[]

Description

Type

object

Required

- **domainPatterns**

Property	Type	Description
domainPatterns	**array (string)**	domainPatterns is a list of domains for which the desired HSTS annotations are required. If domainPatterns is specified and a route is created with a spec.host matching one of the domains, the route must specify the HSTS Policy components described in the matching RequiredHSTSPolicy. The use of wildcards is allowed like this: **.foo.com matches everything under foo.com. foo.com only matches foo.com, so to cover foo.com and everything under it, you must specify *both**.

Property	Type	Description
includeSubDomainsPolicy	**string**	includeSubDomainsPolicy means the HSTS Policy should apply to any subdomains of the host's domain name. Thus, for the host bar.foo.com, if includeSubDomainsPolicy was set to RequireIncludeSubDomains: - the host app.bar.foo.com would inherit the HSTS Policy of bar.foo.com - the host bar.foo.com would inherit the HSTS Policy of bar.foo.com - the host foo.com would NOT inherit the HSTS Policy of bar.foo.com - the host def.foo.com would NOT inherit the HSTS Policy of bar.foo.com
maxAge	**object**	maxAge is the delta time range in seconds during which hosts are regarded as HSTS hosts. If set to 0, it negates the effect, and hosts are removed as HSTS hosts. If set to 0 and includeSubdomains is specified, all subdomains of the host are also removed as HSTS hosts. maxAge is a time-to-live value, and if this policy is not refreshed on a client, the HSTS policy will eventually expire on that client.
namespaceSelector	**object**	namespaceSelector specifies a label selector such that the policy applies only to those routes that are in namespaces with labels that match the selector, and are in one of the DomainPatterns. Defaults to the empty LabelSelector, which matches everything.
preloadPolicy	**string**	preloadPolicy directs the client to include hosts in its host preload list so that it never needs to do an initial load to get the HSTS header (note that this is not defined in RFC 6797 and is therefore client implementation-dependent).

16.1.10. .spec.requiredHSTSPolicies[].maxAge

Description

maxAge is the delta time range in seconds during which hosts are regarded as HSTS hosts. If set to 0, it negates the effect, and hosts are removed as HSTS hosts. If set to 0 and includeSubdomains is specified, all subdomains of the host are also removed as HSTS hosts. maxAge is a time-to-live value, and if this policy is not refreshed on a client, the HSTS policy will eventually expire on that client.

Type

object

Property	Type	Description
largestMaxAge	**integer**	The largest allowed value (in seconds) of the RequiredHSTSPolicy max-age This value can be left unspecified, in which case no upper limit is enforced.
smallestMaxAge	**integer**	The smallest allowed value (in seconds) of the RequiredHSTSPolicy max-age Setting max-age=0 allows the deletion of an existing HSTS header from a host. This is a necessary tool for administrators to quickly correct mistakes. This value can be left unspecified, in which case no lower limit is enforced.

16.1.11. .spec.requiredHSTSPolicies[].namespaceSelector

Description

namespaceSelector specifies a label selector such that the policy applies only to those routes that are in namespaces with labels that match the selector, and are in one of the DomainPatterns. Defaults to the empty LabelSelector, which matches everything.

Type

object

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

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

16.1.12. .spec.requiredHSTSPolicies[].namespaceSelector.matchExpressions

Description

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

Type

array

16.1.13. .spec.requiredHSTSPolicies[].namespaceSelector.matchExpressions[]

Description

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

Type

object

Required

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

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

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

16.1.14. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

Property	Type	Description
componentRoutes	**array**	componentRoutes is where participating operators place the current route status for routes whose hostnames and serving certificates can be customized by the cluster-admin.
componentRoutes[]	**object**	ComponentRouteStatus contains information allowing configuration of a route's hostname and serving certificate.

Property	Type	Description
defaultPlacement	**string**	defaultPlacement is set at installation time to control which nodes will host the ingress router pods by default. The options are control-plane nodes or worker nodes. This field works by dictating how the Cluster Ingress Operator will consider unset replicas and nodePlacement fields in IngressController resources when creating the corresponding Deployments. See the documentation for the IngressController replicas and nodePlacement fields for more information. When omitted, the default value is Workers

16.1.15. .status.componentRoutes

Description

componentRoutes is where participating operators place the current route status for routes whose hostnames and serving certificates can be customized by the cluster-admin.

Type

array

16.1.16. .status.componentRoutes[]

Description

ComponentRouteStatus contains information allowing configuration of a route's hostname and serving certificate.

Type

object

Required

- **defaultHostname**
- **name**
- **namespace**
- **relatedObjects**

Property	Type	Description
conditions	**array**	conditions are used to communicate the state of the componentRoutes entry. Supported conditions include Available, Degraded and Progressing. If available is true, the content served by the route can be accessed by users. This includes cases where a default may continue to serve content while the customized route specified by the cluster-admin is being configured. If Degraded is true, that means something has gone wrong trying to handle the componentRoutes entry. The currentHostnames field may or may not be in effect. If Progressing is true, that means the component is taking some action related to the componentRoutes entry.
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.
consumingUsers	**array (string)**	consumingUsers is a slice of ServiceAccounts that need to have read permission on the servingCertKeyPairSecret secret.
currentHostnames	**array (string)**	currentHostnames is the list of current names used by the route. Typically, this list should consist of a single hostname, but if multiple hostnames are supported by the route the operator may write multiple entries to this list.
defaultHostname	**string**	defaultHostname is the hostname of this route prior to customization.

Property	Type	Description
name	**string**	name is the logical name of the route to customize. It does not have to be the actual name of a route resource but it cannot be renamed. The namespace and name of this componentRoute must match a corresponding entry in the list of spec.componentRoutes if the route is to be customized.
namespace	**string**	namespace is the namespace of the route to customize. It must be a real namespace. Using an actual namespace ensures that no two components will conflict and the same component can be installed multiple times. The namespace and name of this componentRoute must match a corresponding entry in the list of spec.componentRoutes if the route is to be customized.
relatedObjects	**array**	relatedObjects is a list of resources which are useful when debugging or inspecting how spec.componentRoutes is applied.
relatedObjects[]	**object**	ObjectReference contains enough information to let you inspect or modify the referred object.

16.1.17. .status.componentRoutes[].conditions

Description

conditions are used to communicate the state of the componentRoutes entry.
Supported conditions include Available, Degraded and Progressing.

If available is true, the content served by the route can be accessed by users. This includes cases where a default may continue to serve content while the customized route specified by the cluster-admin is being configured.

If Degraded is true, that means something has gone wrong trying to handle the componentRoutes entry. The currentHostnames field may or may not be in effect.

If Progressing is true, that means the component is taking some action related to the componentRoutes entry.

Type

array

16.1.18. .status.componentRoutes[].conditions[]

Description

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

Type

object

Required

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

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

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

16.1.19. .status.componentRoutes[].relatedObjects

Description

relatedObjects is a list of resources which are useful when debugging or inspecting how spec.componentRoutes is applied.

Type

array

16.1.20. .status.componentRoutes[].relatedObjects[]

Description

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

Type

object

Required

- **group**
- **name**
- **resource**

Property	Type	Description
group	**string**	group of the referent.
name	**string**	name of the referent.

Property	Type	Description
namespace	**string**	namespace of the referent.
resource	**string**	resource of the referent.

16.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/ingresses**
  - **DELETE**: delete collection of Ingress
  - **GET**: list objects of kind Ingress
  - **POST**: create an Ingress
- **/apis/config.openshift.io/v1/ingresses/{name}**
  - **DELETE**: delete an Ingress
  - **GET**: read the specified Ingress
  - **PATCH**: partially update the specified Ingress
  - **PUT**: replace the specified Ingress
- **/apis/config.openshift.io/v1/ingresses/{name}/status**
  - **GET**: read status of the specified Ingress
  - **PATCH**: partially update status of the specified Ingress
  - **PUT**: replace status of the specified Ingress

16.2.1. /apis/config.openshift.io/v1/ingresses

HTTP method

DELETE

Description

delete collection of Ingress

Table 16.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Ingress

Table 16.2. HTTP responses

HTTP code	Reponse body
200 - OK	**IngressList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an Ingress

Table 16.3. Query parameters

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

Table 16.4. Body parameters

Parameter	Type	Description
body	**Ingress** schema	

Table 16.5. HTTP responses

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

16.2.2. /apis/config.openshift.io/v1/ingresses/{name}

Table 16.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Ingress

HTTP method

DELETE

Description

delete an Ingress

Table 16.7. 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 16.8. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema

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

HTTP method

GET

Description

read the specified Ingress

Table 16.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Ingress** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Ingress

Table 16.10. Query parameters

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

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

Table 16.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Ingress** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Ingress

Table 16.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 16.13. Body parameters

Parameter	Type	Description
body	**Ingress** schema	

Table 16.14. HTTP responses

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

16.2.3. /apis/config.openshift.io/v1/ingresses/{name}/status

Table 16.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Ingress

HTTP method

GET

Description

read status of the specified Ingress

Table 16.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Ingress** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Ingress

Table 16.17. Query parameters

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

Table 16.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Ingress** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Ingress

Table 16.19. Query parameters

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

Table 16.20. Body parameters

Parameter	Type	Description
body	**Ingress** schema	

Table 16.21. HTTP responses

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

CHAPTER 17. NETWORK [CONFIG.OPENSHIFT.IO/V1]

Description

Network holds cluster-wide information about Network. The canonical name is **cluster**. It is used to configure the desired network configuration, such as: IP address pools for services/pod IPs, network plugin, etc. Please view network.spec for an explanation on what applies when configuring this resource.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

17.1. 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

Property	Type	Description
spec	**object**	spec holds user settable values for configuration. As a general rule, this SHOULD NOT be read directly. Instead, you should consume the NetworkStatus, as it indicates the currently deployed configuration. Currently, most spec fields are immutable after installation. Please view the individual ones for further details on each.
status	**object**	status holds observed values from the cluster. They may not be overridden.

17.1.1. .spec

Description

spec holds user settable values for configuration. As a general rule, this SHOULD NOT be read directly. Instead, you should consume the NetworkStatus, as it indicates the currently deployed configuration. Currently, most spec fields are immutable after installation. Please view the individual ones for further details on each.

Type

object

Property	Type	Description
clusterNetwork	**array**	IP address pool to use for pod IPs. This field is immutable after installation.
clusterNetwork[]	**object**	ClusterNetworkEntry is a contiguous block of IP addresses from which pod IPs are allocated.
externalIP	**object**	externalIP defines configuration for controllers that affect Service.ExternalIP. If nil, then ExternalIP is not allowed to be set.

Property	Type	Description
networkDiagnostics	**object**	networkDiagnostics defines network diagnostics configuration. Takes precedence over spec.disableNetworkDiagnostics in network.operator.openshift.io. If networkDiagnostics is not specified or is empty, and the spec.disableNetworkDiagnostics flag in network.operator.openshift.io is set to true, the network diagnostics feature will be disabled.
networkType	**string**	NetworkType is the plugin that is to be deployed (e.g. OVNKubernetes). This should match a value that the cluster-network-operator understands, or else no networking will be installed. Currently supported values are: - OVNKubernetes This field is immutable after installation.
serviceNetwork	**array (string)**	IP address pool for services. Currently, we only support a single entry here. This field is immutable after installation.
serviceNodePortRange	**string**	The port range allowed for Services of type NodePort. If not specified, the default of 30000-32767 will be used. Such Services without a NodePort specified will have one automatically allocated from this range. This parameter can be updated after the cluster is installed.

17.1.2. .spec.clusterNetwork

Description

IP address pool to use for pod IPs. This field is immutable after installation.

Type

array

17.1.3. .spec.clusterNetwork[]

Description

ClusterNetworkEntry is a contiguous block of IP addresses from which pod IPs are allocated.

Type

object

Property	Type	Description
cidr	**string**	The complete block for pod IPs.
hostPrefix	**integer**	The size (prefix) of block to allocate to each node. If this field is not used by the plugin, it can be left unset.

17.1.4. .spec.externalIP

Description

externalIP defines configuration for controllers that affect Service.ExternalIP. If nil, then ExternalIP is not allowed to be set.

Type

object

Property	Type	Description
autoAssignCIDRs	**array (string)**	autoAssignCIDRs is a list of CIDRs from which to automatically assign Service.ExternalIP. These are assigned when the service is of type LoadBalancer. In general, this is only useful for bare-metal clusters. In Openshift 3.x, this was misleadingly called "IngressIPs". Automatically assigned External IPs are not affected by any ExternalIPPolicy rules. Currently, only one entry may be provided.
policy	**object**	policy is a set of restrictions applied to the ExternalIP field. If nil or empty, then ExternalIP is not allowed to be set.

17.1.5. .spec.externalIP.policy

Description

policy is a set of restrictions applied to the ExternalIP field. If nil or empty, then ExternalIP is not allowed to be set.

Type

object

Property	Type	Description
allowedCIDRs	**array (string)**	allowedCIDRs is the list of allowed CIDRs.
rejectedCIDRs	**array (string)**	rejectedCIDRs is the list of disallowed CIDRs. These take precedence over allowedCIDRs.

17.1.6. .spec.networkDiagnostics

Description

networkDiagnostics defines network diagnostics configuration.
Takes precedence over spec.disableNetworkDiagnostics in network.operator.openshift.io. If networkDiagnostics is not specified or is empty, and the spec.disableNetworkDiagnostics flag in network.operator.openshift.io is set to true, the network diagnostics feature will be disabled.

Type

object

Property	Type	Description
mode	**string**	mode controls the network diagnostics mode When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time. The current default is All.
sourcePlacement	**object**	sourcePlacement controls the scheduling of network diagnostics source deployment See NetworkDiagnosticsSourcePlacement for more details about default values.

Property	Type	Description
targetPlacement	**object**	targetPlacement controls the scheduling of network diagnostics target daemonset See NetworkDiagnosticsTargetPlacement for more details about default values.

17.1.7. .spec.networkDiagnostics.sourcePlacement

Description

sourcePlacement controls the scheduling of network diagnostics source deployment
See NetworkDiagnosticsSourcePlacement for more details about default values.

Type

object

Property	Type	Description
nodeSelector	**object (string)**	nodeSelector is the node selector applied to network diagnostics components When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time. The current default is **kubernetes.io/os: linux**.
tolerations	**array**	tolerations is a list of tolerations applied to network diagnostics components When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time. The current default is an empty list.

Property	Type	Description
tolerations[]	**object**	The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

17.1.8. .spec.networkDiagnostics.sourcePlacement.tolerations

Description

tolerations is a list of tolerations applied to network diagnostics components
When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time. The current default is an empty list.

Type

array

17.1.9. .spec.networkDiagnostics.sourcePlacement.tolerations[]

Description

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

Type

object

Property	Type	Description
effect	**string**	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	**string**	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	**string**	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.

Property	Type	Description
tolerationSeconds	**integer**	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
value	**string**	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

17.1.10. .spec.networkDiagnostics.targetPlacement

Description

targetPlacement controls the scheduling of network diagnostics target daemonset
See NetworkDiagnosticsTargetPlacement for more details about default values.

Type

object

Property	Type	Description
nodeSelector	**object (string)**	nodeSelector is the node selector applied to network diagnostics components When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time. The current default is **kubernetes.io/os: linux**.

Property	Type	Description
tolerations	**array**	tolerations is a list of tolerations applied to network diagnostics components When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time. The current default is **- operator: "Exists"** which means that all taints are tolerated.
tolerations[]	**object**	The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

17.1.11. .spec.networkDiagnostics.targetPlacement.tolerations

Description

tolerations is a list of tolerations applied to network diagnostics components
When omitted, this means the user has no opinion and the platform is left to choose reasonable defaults. These defaults are subject to change over time. The current default is **- operator: "Exists"** which means that all taints are tolerated.

Type

array

17.1.12. .spec.networkDiagnostics.targetPlacement.tolerations[]

Description

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

Type

object

Property	Type	Description
effect	**string**	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.

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

17.1.13. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

Property	Type	Description
clusterNetwork	**array**	IP address pool to use for pod IPs.
clusterNetwork[]	**object**	ClusterNetworkEntry is a contiguous block of IP addresses from which pod IPs are allocated.

Property	Type	Description
clusterNetworkMTU	**integer**	ClusterNetworkMTU is the MTU for inter-pod networking.
conditions	**array**	conditions represents the observations of a network.config current state. Known .status.conditions.type are: "NetworkDiagnosticsAvailable"
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.
migration	**object**	Migration contains the cluster network migration configuration.
networkType	**string**	NetworkType is the plugin that is deployed (e.g. OVNKubernetes).
serviceNetwork	**array (string)**	IP address pool for services. Currently, we only support a single entry here.

17.1.14. .status.clusterNetwork

Description

IP address pool to use for pod IPs.

Type

array

17.1.15. .status.clusterNetwork[]

Description

ClusterNetworkEntry is a contiguous block of IP addresses from which pod IPs are allocated.

Type

object

Property	Type	Description
cidr	**string**	The complete block for pod IPs.

Property	Type	Description
hostPrefix	**integer**	The size (prefix) of block to allocate to each node. If this field is not used by the plugin, it can be left unset.

17.1.16. .status.conditions

Description

conditions represents the observations of a network.config current state. Known .status.conditions.type are: "NetworkDiagnosticsAvailable"

Type

array

17.1.17. .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.

17.1.18. .status.migration

Description

Migration contains the cluster network migration configuration.

Type

object

Property	Type	Description
mtu	**object**	MTU is the MTU configuration that is being deployed.
networkType	**string**	NetworkType is the target plugin that is being deployed. DEPRECATED: network type migration is no longer supported, so this should always be unset.

17.1.19. .status.migration.mtu

Description

MTU is the MTU configuration that is being deployed.

Type

object

Property	Type	Description
machine	**object**	Machine contains MTU migration configuration for the machine's uplink.
network	**object**	Network contains MTU migration configuration for the default network.

17.1.20. .status.migration.mtu.machine

Description

Machine contains MTU migration configuration for the machine's uplink.

Type

object

Property	Type	Description
from	**integer**	From is the MTU to migrate from.
to	**integer**	To is the MTU to migrate to.

17.1.21. .status.migration.mtu.network

Description

Network contains MTU migration configuration for the default network.

Type

object

Property	Type	Description
from	**integer**	From is the MTU to migrate from.
to	**integer**	To is the MTU to migrate to.

17.2. API ENDPOINTS

The following API endpoints are available:

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

17.2.1. /apis/config.openshift.io/v1/networks

HTTP method

DELETE

Description

delete collection of Network

Table 17.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Network

Table 17.2. HTTP responses

HTTP code	Reponse body
200 - OK	**NetworkList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Network

Table 17.3. Query parameters

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

Table 17.4. Body parameters

Parameter	Type	Description
body	**Network** schema	

Table 17.5. HTTP responses

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

17.2.2. /apis/config.openshift.io/v1/networks/{name}

Table 17.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Network

HTTP method

DELETE

Description

delete a Network

Table 17.7. 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 17.8. HTTP responses

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

HTTP method

GET

Description

read the specified Network

Table 17.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Network** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Network

Table 17.10. 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 17.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Network** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Network

Table 17.12. Query parameters

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

Table 17.13. Body parameters

Parameter	Type	Description
body	**Network** schema	

Table 17.14. HTTP responses

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

CHAPTER 18. NODE [CONFIG.OPENSHIFT.IO/V1]

Description

Node holds cluster-wide information about node specific features.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

18.1. 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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	status holds observed values.

18.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
cgroupMode	**string**	CgroupMode determines the cgroups version on the node
workerLatencyProfile	**string**	WorkerLatencyProfile determins the how fast the kubelet is updating the status and corresponding reaction of the cluster

18.1.2. .status

Description

status holds observed values.

Type

object

Property	Type	Description
conditions	**array**	conditions contain the details and the current state of the nodes.config object
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.

18.1.3. .status.conditions

Description

conditions contain the details and the current state of the nodes.config object

Type

array

18.1.4. .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.

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

18.2. API ENDPOINTS

The following API endpoints are available:

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

18.2.1. /apis/config.openshift.io/v1/nodes

HTTP method

DELETE

Description

delete collection of Node

Table 18.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Node

Table 18.2. HTTP responses

HTTP code	Reponse body
200 - OK	**NodeList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Node

Table 18.3. Query parameters

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

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

Table 18.4. Body parameters

Parameter	Type	Description
body	**Node** schema	

Table 18.5. HTTP responses

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

18.2.2. /apis/config.openshift.io/v1/nodes/{name}

Table 18.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Node

HTTP method

DELETE

Description

delete a Node

Table 18.7. 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 18.8. HTTP responses

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

HTTP method

GET

Description

read the specified Node

Table 18.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Node** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Node

Table 18.10. 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 18.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Node** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Node

Table 18.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 18.13. Body parameters

Parameter	Type	Description
body	**Node** schema	

Table 18.14. HTTP responses

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

18.2.3. /apis/config.openshift.io/v1/nodes/{name}/status

Table 18.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Node

HTTP method

GET

Description

read status of the specified Node

Table 18.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Node** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Node

Table 18.17. Query parameters

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

Table 18.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Node** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Node

Table 18.19. Query parameters

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

Table 18.20. Body parameters

Parameter	Type	Description
body	**Node** schema	

Table 18.21. HTTP responses

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

CHAPTER 19. OAUTH [CONFIG.OPENSHIFT.IO/V1]

Description

OAuth holds cluster-wide information about OAuth. The canonical name is **cluster**. It is used to configure the integrated OAuth server. This configuration is only honored when the top level Authentication config has type set to IntegratedOAuth.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

19.1. 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

Property	Type	Description
spec	**object**	spec holds user settable values for configuration
status	**object**	status holds observed values from the cluster. They may not be overridden.

19.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
identityProviders	**array**	identityProviders is an ordered list of ways for a user to identify themselves. When this list is empty, no identities are provisioned for users.
identityProviders[]	**object**	IdentityProvider provides identities for users authenticating using credentials
templates	**object**	templates allow you to customize pages like the login page.
tokenConfig	**object**	tokenConfig contains options for authorization and access tokens

19.1.2. .spec.identityProviders

Description

identityProviders is an ordered list of ways for a user to identify themselves. When this list is empty, no identities are provisioned for users.

Type

array

19.1.3. .spec.identityProviders[]

Description

IdentityProvider provides identities for users authenticating using credentials

Type

object

Property	Type	Description
basicAuth	**object**	basicAuth contains configuration options for the BasicAuth IdP
github	**object**	github enables user authentication using GitHub credentials
gitlab	**object**	gitlab enables user authentication using GitLab credentials
google	**object**	google enables user authentication using Google credentials
htpasswd	**object**	htpasswd enables user authentication using an HTPasswd file to validate credentials
keystone	**object**	keystone enables user authentication using keystone password credentials
ldap	**object**	ldap enables user authentication using LDAP credentials
mappingMethod	**string**	mappingMethod determines how identities from this provider are mapped to users Defaults to "claim"
name	**string**	name is used to qualify the identities returned by this provider. - It MUST be unique and not shared by any other identity provider used - It MUST be a valid path segment: name cannot equal "." or ".." or contain "/" or "%" or ":" Ref: https://godoc.org/github.com/openshift/origin/pkg/user/apis/user/validation#ValidateIdentityProviderName
openID	**object**	openID enables user authentication using OpenID credentials

Property	Type	Description
requestHeader	**object**	requestHeader enables user authentication using request header credentials
type	**string**	type identifies the identity provider type for this entry.

19.1.4. .spec.identityProviders[].basicAuth

Description

basicAuth contains configuration options for the BasicAuth IdP

Type

object

Property	Type	Description
ca	**object**	ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.
tlsClientCert	**object**	tlsClientCert is an optional reference to a secret by name that contains the PEM-encoded TLS client certificate to present when connecting to the server. The key "tls.crt" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. If the specified certificate data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.

Property	Type	Description
tlsClientKey	**object**	tlsClientKey is an optional reference to a secret by name that contains the PEM-encoded TLS private key for the client certificate referenced in tlsClientCert. The key "tls.key" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. If the specified certificate data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.
url	**string**	url is the remote URL to connect to

19.1.5. .spec.identityProviders[].basicAuth.ca

Description

ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

19.1.6. .spec.identityProviders[].basicAuth.tlsClientCert

Description

tlsClientCert is an optional reference to a secret by name that contains the PEM-encoded TLS client certificate to present when connecting to the server. The key "tls.crt" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. If the

specified certificate data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.7. .spec.identityProviders[].basicAuth.tlsClientKey

Description

tlsClientKey is an optional reference to a secret by name that contains the PEM-encoded TLS private key for the client certificate referenced in tlsClientCert. The key "tls.key" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. If the specified certificate data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.8. .spec.identityProviders[].github

Description

github enables user authentication using GitHub credentials

Type

object

Property	Type	Description

Property	Type	Description
ca	**object**	ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. This can only be configured when hostname is set to a non-empty value. The namespace for this config map is openshift-config.
clientID	**string**	clientID is the oauth client ID
clientSecret	**object**	clientSecret is a required reference to the secret by name containing the oauth client secret. The key "clientSecret" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.
hostname	**string**	hostname is the optional domain (e.g. "mycompany.com") for use with a hosted instance of GitHub Enterprise. It must match the GitHub Enterprise settings value configured at /setup/settings#hostname.
organizations	**array (string)**	organizations optionally restricts which organizations are allowed to log in
teams	**array (string)**	teams optionally restricts which teams are allowed to log in. Format is <org>/<team>.

19.1.9. .spec.identityProviders[].github.ca

Description

ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. This can only be configured when hostname is set to a non-empty value. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

19.1.10. .spec.identityProviders[].github.clientSecret

Description

clientSecret is a required reference to the secret by name containing the oauth client secret. The key "clientSecret" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.11. .spec.identityProviders[].gitlab

Description

gitlab enables user authentication using GitLab credentials

Type

object

Property	Type	Description

Property	Type	Description
ca	**object**	ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.
clientID	**string**	clientID is the oauth client ID
clientSecret	**object**	clientSecret is a required reference to the secret by name containing the oauth client secret. The key "clientSecret" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.
url	**string**	url is the oauth server base URL

19.1.12. .spec.identityProviders[].gitlab.ca

Description

ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

19.1.13. .spec.identityProviders[].gitlab.clientSecret

Description

clientSecret is a required reference to the secret by name containing the oauth client secret. The key "clientSecret" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.14. .spec.identityProviders[].google

Description

google enables user authentication using Google credentials

Type

object

Property	Type	Description
clientID	**string**	clientID is the oauth client ID
clientSecret	**object**	clientSecret is a required reference to the secret by name containing the oauth client secret. The key "clientSecret" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.

Property	Type	Description
hostedDomain	**string**	hostedDomain is the optional Google App domain (e.g. "mycompany.com") to restrict logins to

19.1.15. .spec.identityProviders[].google.clientSecret

Description

clientSecret is a required reference to the secret by name containing the oauth client secret. The key "clientSecret" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.16. .spec.identityProviders[].htpasswd

Description

htpasswd enables user authentication using an HTPasswd file to validate credentials

Type

object

Property	Type	Description
fileData	**object**	fileData is a required reference to a secret by name containing the data to use as the htpasswd file. The key "htpasswd" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. If the specified htpasswd data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.

19.1.17. .spec.identityProviders[].htpasswd.fileData

Description

fileData is a required reference to a secret by name containing the data to use as the htpasswd file. The key "htpasswd" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. If the specified htpasswd data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.18. .spec.identityProviders[].keystone

Description

keystone enables user authentication using keystone password credentials

Type

object

Property	Type	Description
ca	**object**	ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.
domainName	**string**	domainName is required for keystone v3

Property	Type	Description
tlsClientCert	**object**	tlsClientCert is an optional reference to a secret by name that contains the PEM-encoded TLS client certificate to present when connecting to the server. The key "tls.crt" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. If the specified certificate data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.
tlsClientKey	**object**	tlsClientKey is an optional reference to a secret by name that contains the PEM-encoded TLS private key for the client certificate referenced in tlsClientCert. The key "tls.key" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. If the specified certificate data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.
url	**string**	url is the remote URL to connect to

19.1.19. .spec.identityProviders[].keystone.ca

Description

ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

19.1.20. .spec.identityProviders[].keystone.tlsClientCert

Description

tlsClientCert is an optional reference to a secret by name that contains the PEM-encoded TLS client certificate to present when connecting to the server. The key "tls.crt" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. If the specified certificate data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.21. .spec.identityProviders[].keystone.tlsClientKey

Description

tlsClientKey is an optional reference to a secret by name that contains the PEM-encoded TLS private key for the client certificate referenced in tlsClientCert. The key "tls.key" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. If the specified certificate data is not valid, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.22. .spec.identityProviders[].ldap

Description

ldap enables user authentication using LDAP credentials

Type

object

Property	Type	Description
attributes	**object**	attributes maps LDAP attributes to identities
bindDN	**string**	bindDN is an optional DN to bind with during the search phase.
bindPassword	**object**	bindPassword is an optional reference to a secret by name containing a password to bind with during the search phase. The key "bindPassword" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.
ca	**object**	ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.

Property	Type	Description
insecure	**boolean**	insecure, if true, indicates the connection should not use TLS WARNING: Should not be set to **true** with the URL scheme "ldaps://" as "ldaps://" URLs always attempt to connect using TLS, even when **insecure** is set to **true** When **true**, "ldap://" URLS connect insecurely. When **false**, "ldap://" URLs are upgraded to a TLS connection using StartTLS as specified in https://tools.ietf.org/html/rfc2830.
url	**string**	url is an RFC 2255 URL which specifies the LDAP search parameters to use. The syntax of the URL is: ldap://host:port/basedn?attribute?scope?filter

19.1.23. .spec.identityProviders[].ldap.attributes

Description

attributes maps LDAP attributes to identities

Type

object

Property	Type	Description
email	**array (string)**	email is the list of attributes whose values should be used as the email address. Optional. If unspecified, no email is set for the identity
id	**array (string)**	id is the list of attributes whose values should be used as the user ID. Required. First non-empty attribute is used. At least one attribute is required. If none of the listed attribute have a value, authentication fails. LDAP standard identity attribute is "dn"

Property	Type	Description
name	**array (string)**	name is the list of attributes whose values should be used as the display name. Optional. If unspecified, no display name is set for the identity LDAP standard display name attribute is "cn"
preferredUsername	**array (string)**	preferredUsername is the list of attributes whose values should be used as the preferred username. LDAP standard login attribute is "uid"

19.1.24. .spec.identityProviders[].ldap.bindPassword

Description

bindPassword is an optional reference to a secret by name containing a password to bind with during the search phase. The key "bindPassword" is used to locate the data. If specified and the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.25. .spec.identityProviders[].ldap.ca

Description

ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

19.1.26. .spec.identityProviders[].openID

Description

openID enables user authentication using OpenID credentials

Type

object

Property	Type	Description
ca	**object**	ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.
claims	**object**	claims mappings
clientID	**string**	clientID is the oauth client ID
clientSecret	**object**	clientSecret is a required reference to the secret by name containing the oauth client secret. The key "clientSecret" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.

Property	Type	Description
extraAuthorizeParameters	**object (string)**	extraAuthorizeParameters are any custom parameters to add to the authorize request.
extraScopes	**array (string)**	extraScopes are any scopes to request in addition to the standard "openid" scope.
issuer	**string**	issuer is the URL that the OpenID Provider asserts as its Issuer Identifier. It must use the https scheme with no query or fragment component.

19.1.27. .spec.identityProviders[].openID.ca

Description

ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca.crt" is used to locate the data. If specified and the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. If empty, the default system roots are used. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

19.1.28. .spec.identityProviders[].openID.claims

Description

claims mappings

Type

object

Property	Type	Description

Property	Type	Description
email	**array (string)**	email is the list of claims whose values should be used as the email address. Optional. If unspecified, no email is set for the identity
groups	**array (string)**	groups is the list of claims value of which should be used to synchronize groups from the OIDC provider to OpenShift for the user. If multiple claims are specified, the first one with a non-empty value is used.
name	**array (string)**	name is the list of claims whose values should be used as the display name. Optional. If unspecified, no display name is set for the identity
preferredUsername	**array (string)**	preferredUsername is the list of claims whose values should be used as the preferred username. If unspecified, the preferred username is determined from the value of the sub claim

19.1.29. .spec.identityProviders[].openID.clientSecret

Description

clientSecret is a required reference to the secret by name containing the oauth client secret. The key "clientSecret" is used to locate the data. If the secret or expected key is not found, the identity provider is not honored. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.30. .spec.identityProviders[].requestHeader

Description

requestHeader enables user authentication using request header credentials

Type

object

Property	Type	Description
ca	**object**	ca is a required reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. Specifically, it allows verification of incoming requests to prevent header spoofing. The key "ca.crt" is used to locate the data. If the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. The namespace for this config map is openshift-config.
challengeURL	**string**	challengeURL is a URL to redirect unauthenticated /authorize requests to Unauthenticated requests from OAuth clients which expect WWW-Authenticate challenges will be redirected here. ${url} is replaced with the current URL, escaped to be safe in a query parameter https://www.example.com/sso-login?then=${url} ${query} is replaced with the current query string https://www.example.com/auth-proxy/oauth/authorize?${query} Required when challenge is set to true.
clientCommonNames	**array (string)**	clientCommonNames is an optional list of common names to require a match from. If empty, any client certificate validated against the clientCA bundle is considered authoritative.

Property	Type	Description
emailHeaders	**array (string)**	emailHeaders is the set of headers to check for the email address
headers	**array (string)**	headers is the set of headers to check for identity information
loginURL	**string**	loginURL is a URL to redirect unauthenticated /authorize requests to Unauthenticated requests from OAuth clients which expect interactive logins will be redirected here ${url} is replaced with the current URL, escaped to be safe in a query parameter https://www.example.com/sso-login?then=${url} ${query} is replaced with the current query string https://www.example.com/auth-proxy/oauth/authorize?${query} Required when login is set to true.
nameHeaders	**array (string)**	nameHeaders is the set of headers to check for the display name
preferredUsernameHeaders	**array (string)**	preferredUsernameHeaders is the set of headers to check for the preferred username

19.1.31. .spec.identityProviders[].requestHeader.ca

Description

ca is a required reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. Specifically, it allows verification of incoming requests to prevent header spoofing. The key "ca.crt" is used to locate the data. If the config map or expected key is not found, the identity provider is not honored. If the specified ca data is not valid, the identity provider is not honored. The namespace for this config map is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

19.1.32. .spec.templates

Description

templates allow you to customize pages like the login page.

Type

object

Property	Type	Description
error	**object**	error is the name of a secret that specifies a go template to use to render error pages during the authentication or grant flow. The key "errors.html" is used to locate the template data. If specified and the secret or expected key is not found, the default error page is used. If the specified template is not valid, the default error page is used. If unspecified, the default error page is used. The namespace for this secret is openshift-config.
login	**object**	login is the name of a secret that specifies a go template to use to render the login page. The key "login.html" is used to locate the template data. If specified and the secret or expected key is not found, the default login page is used. If the specified template is not valid, the default login page is used. If unspecified, the default login page is used. The namespace for this secret is openshift-config.

Property	Type	Description
providerSelection	**object**	providerSelection is the name of a secret that specifies a go template to use to render the provider selection page. The key "providers.html" is used to locate the template data. If specified and the secret or expected key is not found, the default provider selection page is used. If the specified template is not valid, the default provider selection page is used. If unspecified, the default provider selection page is used. The namespace for this secret is openshift-config.

19.1.33. .spec.templates.error

Description

error is the name of a secret that specifies a go template to use to render error pages during the authentication or grant flow. The key "errors.html" is used to locate the template data. If specified and the secret or expected key is not found, the default error page is used. If the specified template is not valid, the default error page is used. If unspecified, the default error page is used. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.34. .spec.templates.login

Description

login is the name of a secret that specifies a go template to use to render the login page. The key "login.html" is used to locate the template data. If specified and the secret or expected key is not found, the default login page is used. If the specified template is not valid, the default login page is used. If unspecified, the default login page is used. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.35. .spec.templates.providerSelection

Description

providerSelection is the name of a secret that specifies a go template to use to render the provider selection page. The key "providers.html" is used to locate the template data. If specified and the secret or expected key is not found, the default provider selection page is used. If the specified template is not valid, the default provider selection page is used. If unspecified, the default provider selection page is used. The namespace for this secret is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

19.1.36. .spec.tokenConfig

Description

tokenConfig contains options for authorization and access tokens

Type

object

Property	Type	Description

Property	Type	Description
accessTokenInactivityTimeout	**string**	accessTokenInactivityTimeout defines the token inactivity timeout for tokens granted by any client. The value represents the maximum amount of time that can occur between consecutive uses of the token. Tokens become invalid if they are not used within this temporal window. The user will need to acquire a new token to regain access once a token times out. Takes valid time duration string such as "5m", "1.5h" or "2h45m". The minimum allowed value for duration is 300s (5 minutes). If the timeout is configured per client, then that value takes precedence. If the timeout value is not specified and the client does not override the value, then tokens are valid until their lifetime. WARNING: existing tokens' timeout will not be affected (lowered) by changing this value
accessTokenInactivityTimeoutSeconds	**integer**	accessTokenInactivityTimeoutSeconds - DEPRECATED: setting this field has no effect.
accessTokenMaxAgeSeconds	**integer**	accessTokenMaxAgeSeconds defines the maximum age of access tokens

19.1.37. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

19.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/oauths**
  - **DELETE**: delete collection of OAuth

- **GET**: list objects of kind OAuth
- **POST**: create an OAuth

- **/apis/config.openshift.io/v1/oauths/{name}**
  - **DELETE**: delete an OAuth
  - **GET**: read the specified OAuth
  - **PATCH**: partially update the specified OAuth
  - **PUT**: replace the specified OAuth
- **/apis/config.openshift.io/v1/oauths/{name}/status**
  - **GET**: read status of the specified OAuth
  - **PATCH**: partially update status of the specified OAuth
  - **PUT**: replace status of the specified OAuth

19.2.1. /apis/config.openshift.io/v1/oauths

HTTP method

DELETE

Description

delete collection of OAuth

Table 19.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind OAuth

Table 19.2. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an OAuth

Table 19.3. Query parameters

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

Table 19.4. Body parameters

Parameter	Type	Description
body	**OAuth** schema	

Table 19.5. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuth** schema
201 - Created	**OAuth** schema
202 - Accepted	**OAuth** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

19.2.2. /apis/config.openshift.io/v1/oauths/{name}

Table 19.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuth

HTTP method

DELETE

Description

delete an OAuth

Table 19.7. 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 19.8. HTTP responses

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

HTTP method

GET

Description

read the specified OAuth

Table 19.9. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuth** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified OAuth

Table 19.10. 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 19.11. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuth** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified OAuth

Table 19.12. Query parameters

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

Table 19.13. Body parameters

Parameter	Type	Description
body	**OAuth** schema	

Table 19.14. HTTP responses

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

19.2.3. /apis/config.openshift.io/v1/oauths/{name}/status

Table 19.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuth

HTTP method

GET

Description

read status of the specified OAuth

Table 19.16. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuth** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified OAuth

Table 19.17. 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 19.18. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuth** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified OAuth

Table 19.19. 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: - 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 19.20. Body parameters

Parameter	Type	Description
body	**OAuth** schema	

Table 19.21. HTTP responses

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

CHAPTER 20. OPERATORHUB [CONFIG.OPENSHIFT.IO/V1]

Description

OperatorHub is the Schema for the operatorhubs API. It can be used to change the state of the default hub sources for OperatorHub on the cluster from enabled to disabled and vice versa. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

20.1. 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**	OperatorHubSpec defines the desired state of OperatorHub

Property	Type	Description
status	**object**	OperatorHubStatus defines the observed state of OperatorHub. The current state of the default hub sources will always be reflected here.

20.1.1. .spec

Description

OperatorHubSpec defines the desired state of OperatorHub

Type

object

Property	Type	Description
disableAllDefaultSources	**boolean**	disableAllDefaultSources allows you to disable all the default hub sources. If this is true, a specific entry in sources can be used to enable a default source. If this is false, a specific entry in sources can be used to disable or enable a default source.
sources	**array**	sources is the list of default hub sources and their configuration. If the list is empty, it implies that the default hub sources are enabled on the cluster unless disableAllDefaultSources is true. If disableAllDefaultSources is true and sources is not empty, the configuration present in sources will take precedence. The list of default hub sources and their current state will always be reflected in the status block.
sources[]	**object**	HubSource is used to specify the hub source and its configuration

20.1.2. .spec.sources

Description

sources is the list of default hub sources and their configuration. If the list is empty, it implies that the default hub sources are enabled on the cluster unless disableAllDefaultSources is true. If disableAllDefaultSources is true and sources is not empty, the configuration present in sources will

take precedence. The list of default hub sources and their current state will always be reflected in the status block.

Type

array

20.1.3. .spec.sources[]

Description

HubSource is used to specify the hub source and its configuration

Type

object

Property	Type	Description
disabled	**boolean**	disabled is used to disable a default hub source on cluster
name	**string**	name is the name of one of the default hub sources

20.1.4. .status

Description

OperatorHubStatus defines the observed state of OperatorHub. The current state of the default hub sources will always be reflected here.

Type

object

Property	Type	Description
sources	**array**	sources encapsulates the result of applying the configuration for each hub source
sources[]	**object**	HubSourceStatus is used to reflect the current state of applying the configuration to a default source

20.1.5. .status.sources

Description

sources encapsulates the result of applying the configuration for each hub source

Type

array

20.1.6. .status.sources[]

Description

HubSourceStatus is used to reflect the current state of applying the configuration to a default source

Type

object

Property	Type	Description
disabled	**boolean**	disabled is used to disable a default hub source on cluster
message	**string**	message provides more information regarding failures
name	**string**	name is the name of one of the default hub sources
status	**string**	status indicates success or failure in applying the configuration

20.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/operatorhubs**
  - **DELETE**: delete collection of OperatorHub
  - **GET**: list objects of kind OperatorHub
  - **POST**: create an OperatorHub
- **/apis/config.openshift.io/v1/operatorhubs/{name}**
  - **DELETE**: delete an OperatorHub
  - **GET**: read the specified OperatorHub
  - **PATCH**: partially update the specified OperatorHub
  - **PUT**: replace the specified OperatorHub
- **/apis/config.openshift.io/v1/operatorhubs/{name}/status**
  - **GET**: read status of the specified OperatorHub
  - **PATCH**: partially update status of the specified OperatorHub
  - **PUT**: replace status of the specified OperatorHub

20.2.1. /apis/config.openshift.io/v1/operatorhubs

HTTP method

DELETE

Description

delete collection of OperatorHub

Table 20.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind OperatorHub

Table 20.2. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorHubList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an OperatorHub

Table 20.3. Query parameters

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

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

Table 20.4. Body parameters

Parameter	Type	Description
body	**OperatorHub** schema	

Table 20.5. HTTP responses

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

20.2.2. /apis/config.openshift.io/v1/operatorhubs/{name}

Table 20.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the OperatorHub

HTTP method

DELETE

Description

delete an OperatorHub

Table 20.7. 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 20.8. HTTP responses

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

HTTP method

GET

Description

read the specified OperatorHub

Table 20.9. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorHub** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified OperatorHub

Table 20.10. 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 20.11. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorHub** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified OperatorHub

Table 20.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 20.13. Body parameters

Parameter	Type	Description
body	**OperatorHub** schema	

Table 20.14. HTTP responses

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

20.2.3. /apis/config.openshift.io/v1/operatorhubs/{name}/status

Table 20.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the OperatorHub

HTTP method

GET

Description

read status of the specified OperatorHub

Table 20.16. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorHub** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified OperatorHub

Table 20.17. Query parameters

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

Table 20.18. HTTP responses

HTTP code	Reponse body
200 - OK	**OperatorHub** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified OperatorHub

Table 20.19. Query parameters

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

Table 20.20. Body parameters

Parameter	Type	Description
body	**OperatorHub** schema	

Table 20.21. HTTP responses

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

CHAPTER 21. PROJECT [CONFIG.OPENSHIFT.IO/V1]

Description

Project holds cluster-wide information about Project. The canonical name is **cluster**

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

Type

object

Required

- **spec**

21.1. 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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

21.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
projectRequestMessage	**string**	projectRequestMessage is the string presented to a user if they are unable to request a project via the projectrequest api endpoint
projectRequestTemplate	**object**	projectRequestTemplate is the template to use for creating projects in response to projectrequest. This must point to a template in 'openshift-config' namespace. It is optional. If it is not specified, a default template is used.

21.1.2. .spec.projectRequestTemplate

Description

projectRequestTemplate is the template to use for creating projects in response to projectrequest. This must point to a template in 'openshift-config' namespace. It is optional. If it is not specified, a default template is used.

Type

object

Property	Type	Description
name	**string**	name is the metadata.name of the referenced project request template

21.1.3. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

21.2. API ENDPOINTS

The following API endpoints are available:

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

21.2.1. /apis/config.openshift.io/v1/projects

HTTP method

DELETE

Description

delete collection of Project

Table 21.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Project

Table 21.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ProjectList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Project

Table 21.3. Query parameters

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

Table 21.4. Body parameters

Parameter	Type	Description
body	**Project** schema	

Table 21.5. HTTP responses

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

21.2.2. /apis/config.openshift.io/v1/projects/{name}

Table 21.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Project

HTTP method

DELETE

Description

delete a Project

Table 21.7. 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 21.8. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema

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

HTTP method

GET

Description

read the specified Project

Table 21.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Project** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Project

Table 21.10. Query parameters

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

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

Table 21.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Project** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Project

Table 21.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 21.13. Body parameters

Parameter	Type	Description
body	**Project** schema	

Table 21.14. HTTP responses

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

21.2.3. /apis/config.openshift.io/v1/projects/{name}/status

Table 21.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Project

HTTP method

GET

Description

read status of the specified Project

Table 21.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Project** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Project

Table 21.17. Query parameters

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

Table 21.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Project** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Project

Table 21.19. Query parameters

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

Table 21.20. Body parameters

Parameter	Type	Description
body	**Project** schema	

Table 21.21. HTTP responses

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

CHAPTER 22. PROJECTHELMCHARTREPOSITORY [HELM.OPENSHIFT.IO/V1BETA1]

Description

ProjectHelmChartRepository holds namespace-wide configuration for proxied Helm chart repository Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

22.1. 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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	Observed status of the repository within the namespace..

22.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description
connectionConfig	**object**	Required configuration for connecting to the chart repo
description	**string**	Optional human readable repository description, it can be used by UI for displaying purposes
disabled	**boolean**	If set to true, disable the repo usage in the namespace
name	**string**	Optional associated human readable repository name, it can be used by UI for displaying purposes

22.1.2. .spec.connectionConfig

Description

Required configuration for connecting to the chart repo

Type

object

Property	Type	Description

Property	Type	Description
basicAuthConfig	**object**	basicAuthConfig is an optional reference to a secret by name that contains the basic authentication credentials to present when connecting to the server. The key "username" is used locate the username. The key "password" is used to locate the password. The namespace for this secret must be same as the namespace where the project helm chart repository is getting instantiated.
ca	**object**	ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca-bundle.crt" is used to locate the data. If empty, the default system roots are used. The namespace for this configmap must be same as the namespace where the project helm chart repository is getting instantiated.
tlsClientConfig	**object**	tlsClientConfig is an optional reference to a secret by name that contains the PEM-encoded TLS client certificate and private key to present when connecting to the server. The key "tls.crt" is used to locate the client certificate. The key "tls.key" is used to locate the private key. The namespace for this secret must be same as the namespace where the project helm chart repository is getting instantiated.
url	**string**	Chart repository URL

22.1.3. .spec.connectionConfig.basicAuthConfig

Description

basicAuthConfig is an optional reference to a secret by name that contains the basic authentication credentials to present when connecting to the server. The key "username" is used locate the username. The key "password" is used to locate the password. The namespace for this secret must

be same as the namespace where the project helm chart repository is getting instantiated.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

22.1.4. .spec.connectionConfig.ca

Description

ca is an optional reference to a config map by name containing the PEM-encoded CA bundle. It is used as a trust anchor to validate the TLS certificate presented by the remote server. The key "ca-bundle.crt" is used to locate the data. If empty, the default system roots are used. The namespace for this configmap must be same as the namespace where the project helm chart repository is getting instantiated.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

22.1.5. .spec.connectionConfig.tlsClientConfig

Description

tlsClientConfig is an optional reference to a secret by name that contains the PEM-encoded TLS client certificate and private key to present when connecting to the server. The key "tls.crt" is used to locate the client certificate. The key "tls.key" is used to locate the private key. The namespace for this secret must be same as the namespace where the project helm chart repository is getting instantiated.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced secret

22.1.6. .status

Description

Observed status of the repository within the namespace..

Type

object

Property	Type	Description
conditions	**array**	conditions is a list of conditions and their statuses
conditions[]	**object**	Condition contains details for one aspect of the current state of this API Resource.

22.1.7. .status.conditions

Description

conditions is a list of conditions and their statuses

Type

array

22.1.8. .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
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.

22.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/helm.openshift.io/v1beta1/projecthelmchartrepositories**
  - **GET**: list objects of kind ProjectHelmChartRepository
- **/apis/helm.openshift.io/v1beta1/namespaces/{namespace}/projecthelmchartrepositories**

- **DELETE**: delete collection of ProjectHelmChartRepository
- **GET**: list objects of kind ProjectHelmChartRepository
- **POST**: create a ProjectHelmChartRepository

- **/apis/helm.openshift.io/v1beta1/namespaces/{namespace}/projecthelmchartrepositories/{name}**
  - **DELETE**: delete a ProjectHelmChartRepository
  - **GET**: read the specified ProjectHelmChartRepository
  - **PATCH**: partially update the specified ProjectHelmChartRepository
  - **PUT**: replace the specified ProjectHelmChartRepository
- **/apis/helm.openshift.io/v1beta1/namespaces/{namespace}/projecthelmchartrepositories/{name}/status**
  - **GET**: read status of the specified ProjectHelmChartRepository
  - **PATCH**: partially update status of the specified ProjectHelmChartRepository
  - **PUT**: replace status of the specified ProjectHelmChartRepository

22.2.1. /apis/helm.openshift.io/v1beta1/projecthelmchartrepositories

HTTP method

GET

Description

list objects of kind ProjectHelmChartRepository

Table 22.1. HTTP responses

HTTP code	Reponse body
200 - OK	**ProjectHelmChartRepositoryList** schema
401 - Unauthorized	Empty

22.2.2. /apis/helm.openshift.io/v1beta1/namespaces/{namespace}/projecthelmchartr

HTTP method

DELETE

Description

delete collection of ProjectHelmChartRepository

Table 22.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind ProjectHelmChartRepository

Table 22.3. HTTP responses

HTTP code	Reponse body
200 - OK	**ProjectHelmChartRepositoryList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a ProjectHelmChartRepository

Table 22.4. Query parameters

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

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

Table 22.5. Body parameters

Parameter	Type	Description
body	**ProjectHelmChartRepository** schema	

Table 22.6. HTTP responses

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

22.2.3. /apis/helm.openshift.io/v1beta1/namespaces/{namespace}/projecthelmchartr

Table 22.7. Global path parameters

Parameter	Type	Description
name	**string**	name of the ProjectHelmChartRepository

HTTP method

DELETE

Description

delete a ProjectHelmChartRepository

Table 22.8. 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 22.9. HTTP responses

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

HTTP method

GET

Description

read the specified ProjectHelmChartRepository

Table 22.10. HTTP responses

HTTP code	Reponse body
200 - OK	**ProjectHelmChartRepository** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ProjectHelmChartRepository

Table 22.11. 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 22.12. HTTP responses

HTTP code	Reponse body
200 - OK	**ProjectHelmChartRepository** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ProjectHelmChartRepository

Table 22.13. 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 22.14. Body parameters

Parameter	Type	Description
body	**ProjectHelmChartRepository** schema	

Table 22.15. HTTP responses

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

22.2.4. /apis/helm.openshift.io/v1beta1/namespaces/{namespace}/projecthelmchartr

Table 22.16. Global path parameters

Parameter	Type	Description
name	**string**	name of the ProjectHelmChartRepository

HTTP method

GET

Description

read status of the specified ProjectHelmChartRepository

Table 22.17. HTTP responses

HTTP code	Reponse body
200 - OK	**ProjectHelmChartRepository** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ProjectHelmChartRepository

Table 22.18. 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 22.19. HTTP responses

HTTP code	Reponse body
200 - OK	**ProjectHelmChartRepository** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ProjectHelmChartRepository

Table 22.20. 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: - 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 22.21. Body parameters

Parameter	Type	Description
body	**ProjectHelmChartRepository** schema	

Table 22.22. HTTP responses

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

CHAPTER 23. PROXY [CONFIG.OPENSHIFT.IO/V1]

Description

Proxy holds cluster-wide information on how to configure default proxies for the cluster. The canonical name is **cluster**

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

Type

object

Required

- **spec**

23.1. 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**	Spec holds user-settable values for the proxy configuration

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

23.1.1. .spec

Description

Spec holds user-settable values for the proxy configuration

Type

object

Property	Type	Description
httpProxy	**string**	httpProxy is the URL of the proxy for HTTP requests. Empty means unset and will not result in an env var.
httpsProxy	**string**	httpsProxy is the URL of the proxy for HTTPS requests. Empty means unset and will not result in an env var.
noProxy	**string**	noProxy is a comma-separated list of hostnames and/or CIDRs and/or IPs for which the proxy should not be used. Empty means unset and will not result in an env var.
readinessEndpoints	**array (string)**	readinessEndpoints is a list of endpoints used to verify readiness of the proxy.

Property	Type	Description
trustedCA	**object**	trustedCA is a reference to a ConfigMap containing a CA certificate bundle. The trustedCA field should only be consumed by a proxy validator. The validator is responsible for reading the certificate bundle from the required key "ca-bundle.crt", merging it with the system default trust bundle, and writing the merged trust bundle to a ConfigMap named "trusted-ca-bundle" in the "openshift-config-managed" namespace. Clients that expect to make proxy connections must use the trusted-ca-bundle for all HTTPS requests to the proxy, and may use the trusted-ca-bundle for non-proxy HTTPS requests as well. The namespace for the ConfigMap referenced by trustedCA is "openshift-config". Here is an example ConfigMap (in yaml): apiVersion: v1 kind: ConfigMap metadata: name: user-ca-bundle namespace: openshift-config data: ca-bundle.crt: | -----BEGIN CERTIFICATE----- Custom CA certificate bundle. -----END CERTIFICATE-----

23.1.2. .spec.trustedCA

Description

trustedCA is a reference to a ConfigMap containing a CA certificate bundle. The trustedCA field should only be consumed by a proxy validator. The validator is responsible for reading the certificate bundle from the required key "ca-bundle.crt", merging it with the system default trust bundle, and writing the merged trust bundle to a ConfigMap named "trusted-ca-bundle" in the "openshift-config-managed" namespace. Clients that expect to make proxy connections must use the trusted-ca-bundle for all HTTPS requests to the proxy, and may use the trusted-ca-bundle for non-proxy HTTPS requests as well.

The namespace for the ConfigMap referenced by trustedCA is "openshift-config". Here is an example ConfigMap (in yaml):

apiVersion: v1 kind: ConfigMap metadata: name: user-ca-bundle namespace: openshift-config data: ca-bundle.crt: \| -----BEGIN CERTIFICATE----- Custom CA certificate bundle. -----END CERTIFICATE-----

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

23.1.3. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

Property	Type	Description
httpProxy	**string**	httpProxy is the URL of the proxy for HTTP requests.
httpsProxy	**string**	httpsProxy is the URL of the proxy for HTTPS requests.
noProxy	**string**	noProxy is a comma-separated list of hostnames and/or CIDRs for which the proxy should not be used.

23.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/config.openshift.io/v1/proxies**
  - **DELETE**: delete collection of Proxy
  - **GET**: list objects of kind Proxy
  - **POST**: create a Proxy
- **/apis/config.openshift.io/v1/proxies/{name}**

- **DELETE**: delete a Proxy
- **GET**: read the specified Proxy
- **PATCH**: partially update the specified Proxy
- **PUT**: replace the specified Proxy

- **/apis/config.openshift.io/v1/proxies/{name}/status**
  - **GET**: read status of the specified Proxy
  - **PATCH**: partially update status of the specified Proxy
  - **PUT**: replace status of the specified Proxy

23.2.1. /apis/config.openshift.io/v1/proxies

HTTP method

DELETE

Description

delete collection of Proxy

Table 23.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Proxy

Table 23.2. HTTP responses

HTTP code	Reponse body
200 - OK	**ProxyList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Proxy

Table 23.3. Query parameters

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

Table 23.4. Body parameters

Parameter	Type	Description
body	**Proxy** schema	

Table 23.5. HTTP responses

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

23.2.2. /apis/config.openshift.io/v1/proxies/{name}

Table 23.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Proxy

HTTP method

DELETE

Description

delete a Proxy

Table 23.7. 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 23.8. HTTP responses

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

HTTP method

GET

Description

read the specified Proxy

Table 23.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Proxy** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Proxy

Table 23.10. 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 23.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Proxy** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Proxy

Table 23.12. Query parameters

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

Table 23.13. Body parameters

Parameter	Type	Description
body	**Proxy** schema	

Table 23.14. HTTP responses

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

23.2.3. /apis/config.openshift.io/v1/proxies/{name}/status

Table 23.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Proxy

HTTP method

GET

Description

read status of the specified Proxy

Table 23.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Proxy** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Proxy

Table 23.17. 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 23.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Proxy** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Proxy

Table 23.19. 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: - 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 23.20. Body parameters

Parameter	Type	Description
body	**Proxy** schema	

Table 23.21. HTTP responses

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

CHAPTER 24. SCHEDULER [CONFIG.OPENSHIFT.IO/V1]

Description

Scheduler holds cluster-wide config information to run the Kubernetes Scheduler and influence its placement decisions. The canonical name for this config is **cluster**.

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

Type

object

Required

- **spec**

24.1. 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**	spec holds user settable values for configuration

Property	Type	Description
status	**object**	status holds observed values from the cluster. They may not be overridden.

24.1.1. .spec

Description

spec holds user settable values for configuration

Type

object

Property	Type	Description

Property	Type	Description
defaultNodeSelector	**string**	defaultNodeSelector helps set the cluster-wide default node selector to restrict pod placement to specific nodes. This is applied to the pods created in all namespaces and creates an intersection with any existing nodeSelectors already set on a pod, additionally constraining that pod's selector. For example, defaultNodeSelector: "type=user-node,region=east" would set nodeSelector field in pod spec to "type=user-node,region=east" to all pods created in all namespaces. Namespaces having project-wide node selectors won't be impacted even if this field is set. This adds an annotation section to the namespace. For example, if a new namespace is created with node-selector='type=user-node,region=east', the annotation openshift.io/node-selector: type=user-node,region=east gets added to the project. When the openshift.io/node-selector annotation is set on the project the value is used in preference to the value we are setting for defaultNodeSelector field. For instance, openshift.io/node-selector: "type=user-node,region=west" means that the default of "type=user-node,region=east" set in defaultNodeSelector would not be applied.

Property	Type	Description
mastersSchedulable	**boolean**	MastersSchedulable allows masters nodes to be schedulable. When this flag is turned on, all the master nodes in the cluster will be made schedulable, so that workload pods can run on them. The default value for this field is false, meaning none of the master nodes are schedulable. Important Note: Once the workload pods start running on the master nodes, extreme care must be taken to ensure that cluster-critical control plane components are not impacted. Please turn on this field after doing due diligence.
policy	**object**	DEPRECATED: the scheduler Policy API has been deprecated and will be removed in a future release. policy is a reference to a ConfigMap containing scheduler policy which has user specified predicates and priorities. If this ConfigMap is not available scheduler will default to use DefaultAlgorithmProvider. The namespace for this configmap is openshift-config.
profile	**string**	profile sets which scheduling profile should be set in order to configure scheduling decisions for new pods. Valid values are "LowNodeUtilization", "HighNodeUtilization", "NoScoring" Defaults to "LowNodeUtilization"

24.1.2. .spec.policy

Description

DEPRECATED: the scheduler Policy API has been deprecated and will be removed in a future release. policy is a reference to a ConfigMap containing scheduler policy which has user specified predicates and priorities. If this ConfigMap is not available scheduler will default to use DefaultAlgorithmProvider. The namespace for this configmap is openshift-config.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the metadata.name of the referenced config map

24.1.3. .status

Description

status holds observed values from the cluster. They may not be overridden.

Type

object

24.2. API ENDPOINTS

The following API endpoints are available:

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

24.2.1. /apis/config.openshift.io/v1/schedulers

HTTP method

DELETE

Description

delete collection of Scheduler

Table 24.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind Scheduler

Table 24.2. HTTP responses

HTTP code	Reponse body
200 - OK	**SchedulerList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a Scheduler

Table 24.3. Query parameters

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

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

Table 24.4. Body parameters

Parameter	Type	Description
body	**Scheduler** schema	

Table 24.5. HTTP responses

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

24.2.2. /apis/config.openshift.io/v1/schedulers/{name}

Table 24.6. Global path parameters

Parameter	Type	Description
name	**string**	name of the Scheduler

HTTP method

DELETE

Description

delete a Scheduler

Table 24.7. 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 24.8. HTTP responses

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

HTTP method

GET

Description

read the specified Scheduler

Table 24.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Scheduler** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Scheduler

Table 24.10. 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 24.11. HTTP responses

HTTP code	Reponse body
200 - OK	**Scheduler** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Scheduler

Table 24.12. Query parameters

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

Parameter	Type	Description
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - 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 24.13. Body parameters

Parameter	Type	Description
body	**Scheduler** schema	

Table 24.14. HTTP responses

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

24.2.3. /apis/config.openshift.io/v1/schedulers/{name}/status

Table 24.15. Global path parameters

Parameter	Type	Description
name	**string**	name of the Scheduler

HTTP method

GET

Description

read status of the specified Scheduler

Table 24.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Scheduler** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Scheduler

Table 24.17. Query parameters

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

Table 24.18. HTTP responses

HTTP code	Reponse body
200 - OK	**Scheduler** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Scheduler

Table 24.19. Query parameters

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

Table 24.20. Body parameters

Parameter	Type	Description
body	**Scheduler** schema	

Table 24.21. HTTP responses

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