OpenShift Container Platform 4.18

Security APIs

Reference guide for security APIs

Last Updated: 2025-05-15

OpenShift Container Platform 4.18 Security APIs

Reference guide for security APIs

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. SECURITY APIS

1.1. CERTIFICATESIGNINGREQUEST [CERTIFICATES.K8S.IO/V1]

Description

CertificateSigningRequest objects provide a mechanism to obtain x509 certificates by submitting a certificate signing request, and having it asynchronously approved and issued.
Kubelets use this API to obtain: 1. client certificates to authenticate to kube-apiserver (with the "kubernetes.io/kube-apiserver-client-kubelet" signerName). 2. serving certificates for TLS endpoints kube-apiserver can connect to securely (with the "kubernetes.io/kubelet-serving" signerName).

This API can be used to request client certificates to authenticate to kube-apiserver (with the "kubernetes.io/kube-apiserver-client" signerName), or to obtain certificates from custom non-Kubernetes signers.

Type

object

1.2. CREDENTIALSREQUEST [CLOUDCREDENTIAL.OPENSHIFT.IO/V1]

Description

CredentialsRequest is the Schema for the credentialsrequests API

Type

object

1.3. PODSECURITYPOLICYREVIEW [SECURITY.OPENSHIFT.IO/V1]

Description

PodSecurityPolicyReview checks which service accounts (not users, since that would be cluster-wide) can create the **PodTemplateSpec** in question.
Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

1.4. PODSECURITYPOLICYSELFSUBJECTREVIEW [SECURITY.OPENSHIFT.IO/V1]

Description

PodSecurityPolicySelfSubjectReview checks whether this user/SA tuple can create the PodTemplateSpec
Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

Type

object

1.5. PODSECURITYPOLICYSUBJECTREVIEW [SECURITY.OPENSHIFT.IO/V1]

Description

PodSecurityPolicySubjectReview checks whether a particular user/SA tuple can create the PodTemplateSpec.

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

Type

object

1.6. RANGEALLOCATION [SECURITY.OPENSHIFT.IO/V1]

Description

RangeAllocation is used so we can easily expose a RangeAllocation typed for security group

Compatibility level 4: No compatibility is provided, the API can change at any point for any reason. These capabilities should not be used by applications needing long term support.

Type

object

1.7. SECRET [V1]

Description

Secret holds secret data of a certain type. The total bytes of the values in the Data field must be less than MaxSecretSize bytes.

Type

object

1.8. SECURITYCONTEXTCONSTRAINTS [SECURITY.OPENSHIFT.IO/V1]

Description

SecurityContextConstraints governs the ability to make requests that affect the SecurityContext that will be applied to a container. For historical reasons SCC was exposed under the core Kubernetes API group. That exposure is deprecated and will be removed in a future release - users should instead use the security.openshift.io group to manage SecurityContextConstraints.

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. SERVICEACCOUNT [V1]

Description

ServiceAccount binds together: * a name, understood by users, and perhaps by peripheral systems, for an identity * a principal that can be authenticated and authorized * a set of secrets

Type

object

CHAPTER 2. CERTIFICATESIGNINGREQUEST [CERTIFICATES.K8S.IO/V1]

Description

CertificateSigningRequest objects provide a mechanism to obtain x509 certificates by submitting a certificate signing request, and having it asynchronously approved and issued.
Kubelets use this API to obtain: 1. client certificates to authenticate to kube-apiserver (with the "kubernetes.io/kube-apiserver-client-kubelet" signerName). 2. serving certificates for TLS endpoints kube-apiserver can connect to securely (with the "kubernetes.io/kubelet-serving" signerName).

This API can be used to request client certificates to authenticate to kube-apiserver (with the "kubernetes.io/kube-apiserver-client" signerName), or to obtain certificates from custom non-Kubernetes signers.

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**	

Property	Type	Description
spec	**object**	CertificateSigningRequestSpec contains the certificate request.
status	**object**	CertificateSigningRequestStatus contains conditions used to indicate approved/denied/failed status of the request, and the issued certificate.

2.1.1. .spec

Description

CertificateSigningRequestSpec contains the certificate request.

Type

object

Required

- **request**
- **signerName**

Property	Type	Description

Property	Type	Description
expirationSeconds	**integer**	expirationSeconds is the requested duration of validity of the issued certificate. The certificate signer may issue a certificate with a different validity duration so a client must check the delta between the notBefore and and notAfter fields in the issued certificate to determine the actual duration. The v1.22+ in-tree implementations of the well-known Kubernetes signers will honor this field as long as the requested duration is not greater than the maximum duration they will honor per the --cluster-signing-duration CLI flag to the Kubernetes controller manager. Certificate signers may not honor this field for various reasons: 1. Old signer that is unaware of the field (such as the in-tree implementations prior to v1.22) 2. Signer whose configured maximum is shorter than the requested duration 3. Signer whose configured minimum is longer than the requested duration The minimum valid value for expirationSeconds is 600, i.e. 10 minutes.
extra	**object**	extra contains extra attributes of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.
extra{}	**array (string)**	

Property	Type	Description
groups	**array (string)**	groups contains group membership of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.
request	**string**	request contains an x509 certificate signing request encoded in a "CERTIFICATE REQUEST" PEM block. When serialized as JSON or YAML, the data is additionally base64-encoded.
signerName	**string**	signerName indicates the requested signer, and is a qualified name. List/watch requests for CertificateSigningRequests can filter on this field using a "spec.signerName=NAME" fieldSelector. Well-known Kubernetes signers are: 1. "kubernetes.io/kube-apiserver-client": issues client certificates that can be used to authenticate to kube-apiserver. Requests for this signer are never auto-approved by kube-controller-manager, can be issued by the "csrsigning" controller in kube-controller-manager. 2. "kubernetes.io/kube-apiserver-client-kubelet": issues client certificates that kubelets use to authenticate to kube-apiserver. Requests for this signer can be auto-approved by the "csrapproving" controller in kube-controller-manager, and can be issued by the "csrsigning" controller in kube-controller-manager. 3. "kubernetes.io/kubelet-serving" issues serving certificates that kubelets use to serve TLS endpoints, which kube-apiserver can connect to securely. Requests for this signer are never auto-approved by kube-controller-

Property	Type	Description
		manager, and can be issued by the "csrsigning" controller in kube-controller-manager. More details are available at https://k8s.io/docs/reference/access-authn-authz/certificate-signing-requests/#kubernetes-signers Custom signerNames can also be specified. The signer defines: 1. Trust distribution: how trust (CA bundles) are distributed. 2. Permitted subjects: and behavior when a disallowed subject is requested. 3. Required, permitted, or forbidden x509 extensions in the request (including whether subjectAltNames are allowed, which types, restrictions on allowed values) and behavior when a disallowed extension is requested. 4. Required, permitted, or forbidden key usages / extended key usages. 5. Expiration/certificate lifetime: whether it is fixed by the signer, configurable by the admin. 6. Whether or not requests for CA certificates are allowed.
uid	**string**	uid contains the uid of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.

Property	Type	Description
usages	**array (string)**	usages specifies a set of key usages requested in the issued certificate. Requests for TLS client certificates typically request: "digital signature", "key encipherment", "client auth". Requests for TLS serving certificates typically request: "key encipherment", "digital signature", "server auth". Valid values are: "signing", "digital signature", "content commitment", "key encipherment", "key agreement", "data encipherment", "cert sign", "crl sign", "encipher only", "decipher only", "any", "server auth", "client auth", "code signing", "email protection", "s/mime", "ipsec end system", "ipsec tunnel", "ipsec user", "timestamping", "ocsp signing", "microsoft sgc", "netscape sgc"
username	**string**	username contains the name of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.

2.1.2. .spec.extra

Description

extra contains extra attributes of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.

Type

object

2.1.3. .status

Description

CertificateSigningRequestStatus contains conditions used to indicate approved/denied/failed status of the request, and the issued certificate.

Type

object

Property	Type	Description
certificate	**string**	certificate is populated with an issued certificate by the signer after an Approved condition is present. This field is set via the /status subresource. Once populated, this field is immutable. If the certificate signing request is denied, a condition of type "Denied" is added and this field remains empty. If the signer cannot issue the certificate, a condition of type "Failed" is added and this field remains empty. Validation requirements: 1. certificate must contain one or more PEM blocks. 2. All PEM blocks must have the "CERTIFICATE" label, contain no headers, and the encoded data must be a BER-encoded ASN.1 Certificate structure as described in section 4 of RFC5280. 3. Non-PEM content may appear before or after the "CERTIFICATE" PEM blocks and is unvalidated, to allow for explanatory text as described in section 5.2 of RFC7468. If more than one PEM block is present, and the definition of the requested spec.signerName does not indicate otherwise, the first block is the issued certificate, and subsequent blocks should be treated as intermediate certificates and presented in TLS handshakes. The certificate is encoded in PEM format. When serialized as JSON or YAML, the data is additionally base64-encoded, so it consists of: base64(-----BEGIN CERTIFICATE----- ... -----END CERTIFICATE-----)

Property	Type	Description
conditions	**array**	conditions applied to the request. Known conditions are "Approved", "Denied", and "Failed".
conditions[]	**object**	CertificateSigningRequestCondition describes a condition of a CertificateSigningRequest object

2.1.4. .status.conditions

Description

conditions applied to the request. Known conditions are "Approved", "Denied", and "Failed".

Type

array

2.1.5. .status.conditions[]

Description

CertificateSigningRequestCondition describes a condition of a CertificateSigningRequest object

Type

object

Required

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

Property	Type	Description
lastTransitionTime	Time	lastTransitionTime is the time the condition last transitioned from one status to another. If unset, when a new condition type is added or an existing condition's status is changed, the server defaults this to the current time.
lastUpdateTime	Time	lastUpdateTime is the time of the last update to this condition
message	**string**	message contains a human readable message with details about the request state

Property	Type	Description
reason	**string**	reason indicates a brief reason for the request state
status	**string**	status of the condition, one of True, False, Unknown. Approved, Denied, and Failed conditions may not be "False" or "Unknown".
type	**string**	type of the condition. Known conditions are "Approved", "Denied", and "Failed". An "Approved" condition is added via the /approval subresource, indicating the request was approved and should be issued by the signer. A "Denied" condition is added via the /approval subresource, indicating the request was denied and should not be issued by the signer. A "Failed" condition is added via the /status subresource, indicating the signer failed to issue the certificate. Approved and Denied conditions are mutually exclusive. Approved, Denied, and Failed conditions cannot be removed once added. Only one condition of a given type is allowed.

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/certificates.k8s.io/v1/certificatesigningrequests**
  - **DELETE**: delete collection of CertificateSigningRequest
  - **GET**: list or watch objects of kind CertificateSigningRequest
  - **POST**: create a CertificateSigningRequest
- **/apis/certificates.k8s.io/v1/watch/certificatesigningrequests**
  - **GET**: watch individual changes to a list of CertificateSigningRequest. deprecated: use the 'watch' parameter with a list operation instead.

'watch' parameter with a list operation instead.

- **/apis/certificates.k8s.io/v1/certificatesigningrequests/{name}**
  - **DELETE**: delete a CertificateSigningRequest
  - **GET**: read the specified CertificateSigningRequest
  - **PATCH**: partially update the specified CertificateSigningRequest
  - **PUT**: replace the specified CertificateSigningRequest
- **/apis/certificates.k8s.io/v1/watch/certificatesigningrequests/{name}**
  - **GET**: watch changes to an object of kind CertificateSigningRequest. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/status**
  - **GET**: read status of the specified CertificateSigningRequest
  - **PATCH**: partially update status of the specified CertificateSigningRequest
  - **PUT**: replace status of the specified CertificateSigningRequest
- **/apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/approval**
  - **GET**: read approval of the specified CertificateSigningRequest
  - **PATCH**: partially update approval of the specified CertificateSigningRequest
  - **PUT**: replace approval of the specified CertificateSigningRequest

2.2.1. /apis/certificates.k8s.io/v1/certificatesigningrequests

HTTP method

DELETE

Description

delete collection of CertificateSigningRequest

Table 2.1. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 2.2. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind CertificateSigningRequest

Table 2.3. HTTP responses

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

HTTP method

POST

Description

create a CertificateSigningRequest

Table 2.4. Query parameters

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

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

Table 2.5. Body parameters

Parameter	Type	Description
body	**CertificateSigningRequest** schema	

Table 2.6. HTTP responses

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

2.2.2. /apis/certificates.k8s.io/v1/watch/certificatesigningrequests

HTTP method

GET

Description

watch individual changes to a list of CertificateSigningRequest. deprecated: use the 'watch' parameter with a list operation instead.

Table 2.7. HTTP responses

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

2.2.3. /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}

Table 2.8. Global path parameters

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

HTTP method

DELETE

Description

delete a CertificateSigningRequest

Table 2.9. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 2.10. HTTP responses

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

HTTP method

GET

Description

read the specified CertificateSigningRequest

Table 2.11. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified CertificateSigningRequest

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. HTTP responses

HTTP code	Reponse body
200 - OK	CertificateSigningRequest schema

HTTP code	Reponse body
201 - Created	**CertificateSigningRequest** schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified CertificateSigningRequest

Table 2.14. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 2.15. Body parameters

Parameter	Type	Description
body	**CertificateSigningRequest** schema	

Table 2.16. HTTP responses

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

2.2.4. /apis/certificates.k8s.io/v1/watch/certificatesigningrequests/{name}

Table 2.17. Global path parameters

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

HTTP method

GET

Description

watch changes to an object of kind CertificateSigningRequest. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 2.18. HTTP responses

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

2.2.5. /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/status

Table 2.19. Global path parameters

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

HTTP method

GET

Description

read status of the specified CertificateSigningRequest

Table 2.20. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified CertificateSigningRequest

Table 2.21. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 2.22. HTTP responses

HTTP code	Reponse body
200 - OK	**CertificateSigningRequest** schema
201 - Created	**CertificateSigningRequest** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified CertificateSigningRequest

Table 2.23. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 2.24. Body parameters

Parameter	Type	Description
body	**CertificateSigningRequest** schema	

Table 2.25. HTTP responses

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

2.2.6. /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/approval

Table 2.26. Global path parameters

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

HTTP method

GET

Description

read approval of the specified CertificateSigningRequest

Table 2.27. HTTP responses

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

HTTP method

PATCH

Description

partially update approval of the specified CertificateSigningRequest

Table 2.28. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 2.29. HTTP responses

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

HTTP method

PUT

Description

replace approval of the specified CertificateSigningRequest

Table 2.30. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 2.31. Body parameters

Parameter	Type	Description
body	**CertificateSigningRequest** schema	

Table 2.32. HTTP responses

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

CHAPTER 3. CREDENTIALSREQUEST [CLOUDCREDENTIAL.OPENSHIFT.IO/V1]

Description

CredentialsRequest is the Schema for the credentialsrequests API

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**	CredentialsRequestSpec defines the desired state of CredentialsRequest

Property	Type	Description
status	**object**	CredentialsRequestStatus defines the observed state of CredentialsRequest

3.1.1. .spec

Description

CredentialsRequestSpec defines the desired state of CredentialsRequest

Type

object

Required

- **secretRef**

Property	Type	Description
cloudTokenPath	**string**	cloudTokenPath is the path where the Kubernetes ServiceAccount token (JSON Web Token) is mounted on the deployment for the workload requesting a credentials secret. The presence of this field in combination with fields such as spec.providerSpec.stsIAMRoleARN indicate that CCO should broker creation of a credentials secret containing fields necessary for token based authentication methods such as with the AWS Secure Token Service (STS). cloudTokenPath may also be used to specify the azure_federated_token_file path used in Azure configuration secrets generated by ccoctl. Defaults to "/var/run/secrets/openshift/serviceaccount/token".
providerSpec	` `	ProviderSpec contains the cloud provider specific credentials specification.
secretRef	**object**	SecretRef points to the secret where the credentials should be stored once generated.

Property	Type	Description
serviceAccountNames	**array (string)**	ServiceAccountNames contains a list of ServiceAccounts that will use permissions associated with this CredentialsRequest. This is not used by CCO, but the information is needed for being able to properly set up access control in the cloud provider when the ServiceAccounts are used as part of the cloud credentials flow.

3.1.2. .spec.secretRef

Description

SecretRef points to the secret where the credentials should be stored once generated.

Type

object

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

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

3.1.3. .status

Description

CredentialsRequestStatus defines the observed state of CredentialsRequest

Type

object

Required

- **lastSyncGeneration**
- **provisioned**

Property	Type	Description
conditions	**array**	Conditions includes detailed status for the CredentialsRequest

Property	Type	Description
conditions[]	**object**	CredentialsRequestCondition contains details for any of the conditions on a CredentialsRequest object
lastSyncCloudCredsSecretResourceVersion	**string**	LastSyncCloudCredsSecretResourceVersion is the resource version of the cloud credentials secret resource when the credentials request resource was last synced. Used to determine if the cloud credentials have been updated since the last sync.
lastSyncGeneration	**integer**	LastSyncGeneration is the generation of the credentials request resource that was last synced. Used to determine if the object has changed and requires a sync.
lastSyncInfrastructureResourceVersion	**string**	LastSyncInfrastructureResourceVersion is the resource version of the Infrastructure resource. It is used to determine if the user provided tags have been updated since the last sync.
lastSyncTimestamp	**string**	LastSyncTimestamp is the time that the credentials were last synced.
providerStatus	``	ProviderStatus contains cloud provider specific status.
provisioned	**boolean**	Provisioned is true once the credentials have been initially provisioned.

3.1.4. .status.conditions

Description

Conditions includes detailed status for the CredentialsRequest

Type

array

3.1.5. .status.conditions[]

Description

CredentialsRequestCondition contains details for any of the conditions on a CredentialsRequest object

Type

object

Required

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

Property	Type	Description
lastProbeTime	**string**	LastProbeTime is the last time we probed the condition
lastTransitionTime	**string**	LastTransitionTime is the last time the condition transitioned from one status to another.
message	**string**	Message is a human-readable message indicating details about the last transition
reason	**string**	Reason is a unique, one-word, CamelCase reason for the condition’s last transition
status	**string**	Status is the status of the condition
type	**string**	Type is the specific type of the condition

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/cloudcredential.openshift.io/v1/credentialsrequests**
  - **GET**: list objects of kind CredentialsRequest
- **/apis/cloudcredential.openshift.io/v1/namespaces/{namespace}/credentialsrequests**
  - **DELETE**: delete collection of CredentialsRequest
  - **GET**: list objects of kind CredentialsRequest
  - **POST**: create a CredentialsRequest

- **/apis/cloudcredential.openshift.io/v1/namespaces/{namespace}/credentialsrequests/{name}**
  - **DELETE**: delete a CredentialsRequest
  - **GET**: read the specified CredentialsRequest
  - **PATCH**: partially update the specified CredentialsRequest
  - **PUT**: replace the specified CredentialsRequest
- **/apis/cloudcredential.openshift.io/v1/namespaces/{namespace}/credentialsrequests/{name}/status**
  - **GET**: read status of the specified CredentialsRequest
  - **PATCH**: partially update status of the specified CredentialsRequest
  - **PUT**: replace status of the specified CredentialsRequest

3.2.1. /apis/cloudcredential.openshift.io/v1/credentialsrequests

HTTP method

GET

Description

list objects of kind CredentialsRequest

Table 3.1. HTTP responses

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

3.2.2. /apis/cloudcredential.openshift.io/v1/namespaces/{namespace}/credentialsrec

HTTP method

DELETE

Description

delete collection of CredentialsRequest

Table 3.2. HTTP responses

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

HTTP method

GET

Description

list objects of kind CredentialsRequest

Table 3.3. HTTP responses

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

HTTP method

POST

Description

create a CredentialsRequest

Table 3.4. Query parameters

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

Table 3.5. Body parameters

Parameter	Type	Description
body	**CredentialsRequest** schema	

Table 3.6. HTTP responses

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

3.2.3. /apis/cloudcredential.openshift.io/v1/namespaces/{namespace}/credentialsrec

Table 3.7. Global path parameters

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

HTTP method

DELETE

Description

delete a CredentialsRequest

Table 3.8. Query parameters

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

Table 3.9. HTTP responses

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

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

HTTP method

GET

Description

read the specified CredentialsRequest

Table 3.10. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified CredentialsRequest

Table 3.11. Query parameters

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

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 3.12. HTTP responses

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

HTTP method

PUT

Description

replace the specified CredentialsRequest

Table 3.13. Query parameters

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

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 3.14. Body parameters

Parameter	Type	Description
body	**CredentialsRequest** schema	

Table 3.15. HTTP responses

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

3.2.4. /apis/cloudcredential.openshift.io/v1/namespaces/{namespace}/credentialsrec

Table 3.16. Global path parameters

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

HTTP method

GET

Description

read status of the specified CredentialsRequest

Table 3.17. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified CredentialsRequest

Table 3.18. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	**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 3.19. HTTP responses

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

HTTP method

PUT

Description

replace status of the specified CredentialsRequest

Table 3.20. Query parameters

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

Parameter	Type	Description
body	CredentialsRequest schema	

Table 3.22. HTTP responses

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

CHAPTER 4. PODSECURITYPOLICYREVIEW [SECURITY.OPENSHIFT.IO/V1]

Description

PodSecurityPolicyReview checks which service accounts (not users, since that would be cluster-wide) can create the **PodTemplateSpec** in question.
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**

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
spec	**object**	PodSecurityPolicyReviewSpec defines specification for PodSecurityPolicyReview
status	**object**	PodSecurityPolicyReviewStatus represents the status of PodSecurityPolicyReview.

4.1.1. .spec

Description

PodSecurityPolicyReviewSpec defines specification for PodSecurityPolicyReview

Type

object

Required

- **template**

Property	Type	Description
serviceAccountNames	**array (string)**	serviceAccountNames is an optional set of ServiceAccounts to run the check with. If serviceAccountNames is empty, the template.spec.serviceAccountName is used, unless it's empty, in which case "default" is used instead. If serviceAccountNames is specified, template.spec.serviceAccountName is ignored.
template	**PodTemplateSpec**	template is the PodTemplateSpec to check. The template.spec.serviceAccountName field is used if serviceAccountNames is empty, unless the template.spec.serviceAccountName is empty, in which case "default" is used. If serviceAccountNames is specified, template.spec.serviceAccountName is ignored.

4.1.2. .status

Description

PodSecurityPolicyReviewStatus represents the status of PodSecurityPolicyReview.

Type

object

Required

- **allowedServiceAccounts**

Property	Type	Description
allowedServiceAccounts	**array**	allowedServiceAccounts returns the list of service accounts in **this** namespace that have the power to create the PodTemplateSpec.
allowedServiceAccounts[]	**object**	ServiceAccountPodSecurityPolicyReviewStatus represents ServiceAccount name and related review status

4.1.3. .status.allowedServiceAccounts

Description

allowedServiceAccounts returns the list of service accounts in **this** namespace that have the power to create the PodTemplateSpec.

Type

array

4.1.4. .status.allowedServiceAccounts[]

Description

ServiceAccountPodSecurityPolicyReviewStatus represents ServiceAccount name and related review status

Type

object

Required

- **name**

Property	Type	Description
allowedBy	**ObjectReference**	allowedBy is a reference to the rule that allows the PodTemplateSpec. A rule can be a SecurityContextConstraint or a PodSecurityPolicy A **nil**, indicates that it was denied.
name	**string**	name contains the allowed and the denied ServiceAccount name
reason	**string**	A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available.

Property	Type	Description
template	PodTemplateSpec	template is the PodTemplateSpec after the defaulting is applied.

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/security.openshift.io/v1/namespaces/{namespace}/podsecuritypolicyreviews**
  - **POST**: create a PodSecurityPolicyReview

4.2.1. /apis/security.openshift.io/v1/namespaces/{namespace}/podsecuritypolicyrevi

Table 4.1. Global query parameters

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

HTTP method

POST

Description

create a PodSecurityPolicyReview

Table 4.2. Body parameters

Parameter	Type	Description
body	PodSecurityPolicyReview schema	

Table 4.3. HTTP responses

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

CHAPTER 5. PODSECURITYPOLICYSELFSUBJECTREVIEW [SECURITY.OPENSHIFT.IO/V1]

Description

PodSecurityPolicySelfSubjectReview checks whether this user/SA tuple can create the PodTemplateSpec
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**

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
spec	**object**	PodSecurityPolicySelfSubjectReviewSpec contains specification for PodSecurityPolicySelfSubjectReview.

Property	Type	Description
status	**object**	PodSecurityPolicySubjectReview Status contains information/status for PodSecurityPolicySubjectReview.

5.1.1. .spec

Description

PodSecurityPolicySelfSubjectReviewSpec contains specification for PodSecurityPolicySelfSubjectReview.

Type

object

Required

- **template**

Property	Type	Description
template	**PodTemplateSpec**	template is the PodTemplateSpec to check.

5.1.2. .status

Description

PodSecurityPolicySubjectReviewStatus contains information/status for PodSecurityPolicySubjectReview.

Type

object

Property	Type	Description
allowedBy	**ObjectReference**	allowedBy is a reference to the rule that allows the PodTemplateSpec. A rule can be a SecurityContextConstraint or a PodSecurityPolicy A **nil**, indicates that it was denied.
reason	**string**	A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available.

Property	Type	Description
template	PodTemplateSpec	template is the PodTemplateSpec after the defaulting is applied.

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/security.openshift.io/v1/namespaces/{namespace}/podsecuritypolicyselfsubjectreviews**
  - **POST**: create a PodSecurityPolicySelfSubjectReview

5.2.1. /apis/security.openshift.io/v1/namespaces/{namespace}/podsecuritypolicyselfs

Table 5.1. Global query parameters

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

HTTP method

POST

Description

create a PodSecurityPolicySelfSubjectReview

Table 5.2. Body parameters

Parameter	Type	Description
body	**PodSecurityPolicySelfSubjectReview** schema	

Table 5.3. HTTP responses

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

CHAPTER 6. PODSECURITYPOLICYSUBJECTREVIEW [SECURITY.OPENSHIFT.IO/V1]

Description

PodSecurityPolicySubjectReview checks whether a particular user/SA tuple can create the PodTemplateSpec.
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**

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
spec	**object**	PodSecurityPolicySubjectReviewSpec defines specification for PodSecurityPolicySubjectReview
status	**object**	PodSecurityPolicySubjectReviewStatus contains information/status for PodSecurityPolicySubjectReview.

6.1.1. .spec

Description

PodSecurityPolicySubjectReviewSpec defines specification for PodSecurityPolicySubjectReview

Type

object

Required

- **template**

Property	Type	Description
groups	**array (string)**	groups is the groups you're testing for.
template	**PodTemplateSpec**	template is the PodTemplateSpec to check. If template.spec.serviceAccountName is empty it will not be defaulted. If its non-empty, it will be checked.
user	**string**	user is the user you're testing for. If you specify "user" but not "group", then is it interpreted as "What if user were not a member of any groups. If user and groups are empty, then the check is performed using **only** the serviceAccountName in the template.

6.1.2. .status

Description

PodSecurityPolicySubjectReviewStatus contains information/status for PodSecurityPolicySubjectReview.

Type

object

Property	Type	Description
allowedBy	**ObjectReference**	allowedBy is a reference to the rule that allows the PodTemplateSpec. A rule can be a SecurityContextConstraint or a PodSecurityPolicy A **nil**, indicates that it was denied.

Property	Type	Description
reason	**string**	A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available.
template	PodTemplateSpec	template is the PodTemplateSpec after the defaulting is applied.

6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/security.openshift.io/v1/namespaces/{namespace}/podsecuritypolicysubjectreviews**
  - **POST**: create a PodSecurityPolicySubjectReview

6.2.1. /apis/security.openshift.io/v1/namespaces/{namespace}/podsecuritypolicysubj

Table 6.1. Global query parameters

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

HTTP method

POST

Description

create a PodSecurityPolicySubjectReview

Table 6.2. Body parameters

Parameter	Type	Description
body	**PodSecurityPolicySubjectReview** schema	

Table 6.3. HTTP responses

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

CHAPTER 7. RANGEALLOCATION [SECURITY.OPENSHIFT.IO/V1]

Description

RangeAllocation is used so we can easily expose a RangeAllocation typed for security group Compatibility level 4: No compatibility is provided, the API can change at any point for any reason. These capabilities should not be used by applications needing long term support.

Type

object

Required

- **range**
- **data**

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
data	**string**	data is a byte array representing the serialized state of a range allocation. It is a bitmap with each bit set to one to represent a range is taken.
kind	**string**	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	**ObjectMeta_v2**	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
range	**string**	range is a string representing a unique label for a range of uids, "1000000000-2000000000/10000".

7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/security.openshift.io/v1/rangeallocations**
  - **DELETE**: delete collection of RangeAllocation
  - **GET**: list or watch objects of kind RangeAllocation
  - **POST**: create a RangeAllocation
- **/apis/security.openshift.io/v1/watch/rangeallocations**
  - **GET**: watch individual changes to a list of RangeAllocation. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/security.openshift.io/v1/rangeallocations/{name}**
  - **DELETE**: delete a RangeAllocation
  - **GET**: read the specified RangeAllocation
  - **PATCH**: partially update the specified RangeAllocation
  - **PUT**: replace the specified RangeAllocation
- **/apis/security.openshift.io/v1/watch/rangeallocations/{name}**
  - **GET**: watch changes to an object of kind RangeAllocation. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

7.2.1. /apis/security.openshift.io/v1/rangeallocations

HTTP method

DELETE

Description

delete collection of RangeAllocation

Table 7.1. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 7.2. HTTP responses

HTTP code	Reponse body
200 - OK	**Status_v8** schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind RangeAllocation

Table 7.3. HTTP responses

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

HTTP method

POST

Description

create a RangeAllocation

Table 7.4. Query parameters

Parameter	Type	Description

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

Table 7.5. Body parameters

Parameter	Type	Description
body	**RangeAllocation** schema	

Table 7.6. HTTP responses

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

7.2.2. /apis/security.openshift.io/v1/watch/rangeallocations

HTTP method

GET

Description

watch individual changes to a list of RangeAllocation. deprecated: use the 'watch' parameter with a list operation instead.

Table 7.7. HTTP responses

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

7.2.3. /apis/security.openshift.io/v1/rangeallocations/{name}

Table 7.8. Global path parameters

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

HTTP method

DELETE

Description

delete a RangeAllocation

Table 7.9. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 7.10. HTTP responses

HTTP code	Reponse body
200 - OK	**Status_v8** schema
202 - Accepted	**Status_v8** schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified RangeAllocation

Table 7.11. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified RangeAllocation

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
fieldValidation	**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. HTTP responses

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

HTTP method

PUT

Description

replace the specified RangeAllocation

Table 7.14. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 7.15. Body parameters

Parameter	Type	Description
body	**RangeAllocation** schema	

Table 7.16. HTTP responses

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

7.2.4. /apis/security.openshift.io/v1/watch/rangeallocations/{name}

Table 7.17. Global path parameters

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

HTTP method

GET

Description

watch changes to an object of kind RangeAllocation. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 7.18. HTTP responses

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

CHAPTER 8. SECRET [V1]

Description

Secret holds secret data of a certain type. The total bytes of the values in the Data field must be less than MaxSecretSize bytes.

Type

object

8.1. SPECIFICATION

Property	Type	Description
apiVersion	**string**	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
data	**object (string)**	Data contains the secret data. Each key must consist of alphanumeric characters, '-', '_' or '.'. The serialized form of the secret data is a base64 encoded string, representing the arbitrary (possibly non-string) data value here. Described in https://tools.ietf.org/html/rfc4648#section-4
immutable	**boolean**	Immutable, if set to true, ensures that data stored in the Secret cannot be updated (only object metadata can be modified). If not set to true, the field can be modified at any time. Defaulted to nil.

Property	Type	Description
kind	**string**	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
stringData	**object (string)**	stringData allows specifying non-binary secret data in string form. It is provided as a write-only input field for convenience. All keys and values are merged into the data field on write, overwriting any existing values. The stringData field is never output when reading from the API.
type	**string**	Used to facilitate programmatic handling of secret data. More info: https://kubernetes.io/docs/concepts/configuration/secret/#secret-types

8.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/secrets**
  - **GET**: list or watch objects of kind Secret
- **/api/v1/watch/secrets**
  - **GET**: watch individual changes to a list of Secret. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/secrets**
  - **DELETE**: delete collection of Secret

  - **GET**: list or watch objects of kind Secret
  - **POST**: create a Secret
- **/api/v1/watch/namespaces/{namespace}/secrets**
  - **GET**: watch individual changes to a list of Secret. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/secrets/{name}**
  - **DELETE**: delete a Secret
  - **GET**: read the specified Secret
  - **PATCH**: partially update the specified Secret
  - **PUT**: replace the specified Secret
- **/api/v1/watch/namespaces/{namespace}/secrets/{name}**
  - **GET**: watch changes to an object of kind Secret. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

8.2.1. /api/v1/secrets

HTTP method

GET

Description

list or watch objects of kind Secret

Table 8.1. HTTP responses

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

8.2.2. /api/v1/watch/secrets

HTTP method

GET

Description

watch individual changes to a list of Secret. deprecated: use the 'watch' parameter with a list operation instead.

Table 8.2. HTTP responses

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

HTTP code	Reponse body
401 - Unauthorized	Empty

8.2.3. /api/v1/namespaces/{namespace}/secrets

HTTP method

DELETE

Description

delete collection of Secret

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

Table 8.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind Secret

Table 8.5. HTTP responses

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

HTTP method

POST

Description

create a Secret

Table 8.6. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 8.7. Body parameters

Parameter	Type	Description
body	**Secret** schema	

Table 8.8. HTTP responses

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

8.2.4. /api/v1/watch/namespaces/{namespace}/secrets

HTTP method

GET

Description

watch individual changes to a list of Secret. deprecated: use the 'watch' parameter with a list operation instead.

Table 8.9. HTTP responses

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

8.2.5. /api/v1/namespaces/{namespace}/secrets/{name}

Table 8.10. Global path parameters

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

HTTP method

DELETE

Description

delete a Secret

Table 8.11. Query parameters

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

Table 8.12. HTTP responses

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

HTTP method

GET

Description

read the specified Secret

Table 8.13. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified Secret

Table 8.14. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 8.15. HTTP responses

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

HTTP method

PUT

Description

replace the specified Secret

Table 8.16. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 8.17. Body parameters

Parameter	Type	Description
body	**Secret** schema	

Table 8.18. HTTP responses

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

8.2.6. /api/v1/watch/namespaces/{namespace}/secrets/{name}

Table 8.19. Global path parameters

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

HTTP method

GET

Description

watch changes to an object of kind Secret. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 8.20. HTTP responses

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

CHAPTER 9. SECURITYCONTEXTCONSTRAINTS [SECURITY.OPENSHIFT.IO/V1]

Description

SecurityContextConstraints governs the ability to make requests that affect the SecurityContext that will be applied to a container. For historical reasons SCC was exposed under the core Kubernetes API group. That exposure is deprecated and will be removed in a future release - users should instead use the security.openshift.io group to manage SecurityContextConstraints.

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

Type

object

Required

- **allowHostDirVolumePlugin**
- **allowHostIPC**
- **allowHostNetwork**
- **allowHostPID**
- **allowHostPorts**
- **allowPrivilegedContainer**
- **readOnlyRootFilesystem**

9.1. SPECIFICATION

Property	Type	Description
allowHostDirVolumePlugin	**boolean**	AllowHostDirVolumePlugin determines if the policy allow containers to use the HostDir volume plugin
allowHostIPC	**boolean**	AllowHostIPC determines if the policy allows host ipc in the containers.
allowHostNetwork	**boolean**	AllowHostNetwork determines if the policy allows the use of HostNetwork in the pod spec.
allowHostPID	**boolean**	AllowHostPID determines if the policy allows host pid in the containers.

Property	Type	Description
allowHostPorts	**boolean**	AllowHostPorts determines if the policy allows host ports in the containers.
allowPrivilegeEscalation	``	AllowPrivilegeEscalation determines if a pod can request to allow privilege escalation. If unspecified, defaults to true.
allowPrivilegedContainer	**boolean**	AllowPrivilegedContainer determines if a container can request to be run as privileged.
allowedCapabilities	``	AllowedCapabilities is a list of capabilities that can be requested to add to the container. Capabilities in this field maybe added at the pod author's discretion. You must not list a capability in both AllowedCapabilities and RequiredDropCapabilities. To allow all capabilities you may use '*'.
allowedFlexVolumes	``	AllowedFlexVolumes is a whitelist of allowed Flexvolumes. Empty or nil indicates that all Flexvolumes may be used. This parameter is effective only when the usage of the Flexvolumes is allowed in the "Volumes" field.
allowedUnsafeSysctls	``	AllowedUnsafeSysctls is a list of explicitly allowed unsafe sysctls, defaults to none. Each entry is either a plain sysctl name or ends in "*" in which case it is considered as a prefix of allowed sysctls. Single * means all unsafe sysctls are allowed. Kubelet has to whitelist all allowed unsafe sysctls explicitly to avoid rejection. Examples: e.g. "foo/*" allows "foo/bar", "foo/baz", etc. e.g. "foo.*" allows "foo.bar", "foo.baz", etc.

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
defaultAddCapabilities	` `	DefaultAddCapabilities is the default set of capabilities that will be added to the container unless the pod spec specifically drops the capability. You may not list a capabiility in both DefaultAddCapabilities and RequiredDropCapabilities.
defaultAllowPrivilegeEscalation	` `	DefaultAllowPrivilegeEscalation controls the default setting for whether a process can gain more privileges than its parent process.
forbiddenSysctls	` `	ForbiddenSysctls is a list of explicitly forbidden sysctls, defaults to none. Each entry is either a plain sysctl name or ends in "*" in which case it is considered as a prefix of forbidden sysctls. Single * means all sysctls are forbidden. Examples: e.g. "foo/*" forbids "foo/bar", "foo/baz", etc. e.g. "foo.*" forbids "foo.bar", "foo.baz", etc.
fsGroup	` `	FSGroup is the strategy that will dictate what fs group is used by the SecurityContext.
groups	` `	The groups that have permission to use this security context constraints

Property	Type	Description
kind	**string**	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
priority	``	Priority influences the sort order of SCCs when evaluating which SCCs to try first for a given pod request based on access in the Users and Groups fields. The higher the int, the higher priority. An unset value is considered a 0 priority. If scores for multiple SCCs are equal they will be sorted from most restrictive to least restrictive. If both priorities and restrictions are equal the SCCs will be sorted by name.
readOnlyRootFilesystem	**boolean**	ReadOnlyRootFilesystem when set to true will force containers to run with a read only root file system. If the container specifically requests to run with a non-read only root file system the SCC should deny the pod. If set to false the container may run with a read only root file system if it wishes but it will not be forced to.

Property	Type	Description
requiredDropCapabilities	``	RequiredDropCapabilities are the capabilities that will be dropped from the container. These are required to be dropped and cannot be added.
runAsUser	``	RunAsUser is the strategy that will dictate what RunAsUser is used in the SecurityContext.
seLinuxContext	``	SELinuxContext is the strategy that will dictate what labels will be set in the SecurityContext.
seccompProfiles	``	SeccompProfiles lists the allowed profiles that may be set for the pod or container's seccomp annotations. An unset (nil) or empty value means that no profiles may be specifid by the pod or container. The wildcard '*' may be used to allow all profiles. When used to generate a value for a pod the first non-wildcard profile will be used as the default.
supplementalGroups	``	SupplementalGroups is the strategy that will dictate what supplemental groups are used by the SecurityContext.
users	``	The users who have permissions to use this security context constraints
volumes	``	Volumes is a white list of allowed volume plugins. FSType corresponds directly with the field names of a VolumeSource (azureFile, configMap, emptyDir). To allow all volumes you may use "*". To allow no volumes, set to ["none"].

9.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/security.openshift.io/v1/securitycontextconstraints**
  - **DELETE**: delete collection of SecurityContextConstraints
  - **GET**: list objects of kind SecurityContextConstraints
  - **POST**: create SecurityContextConstraints
- **/apis/security.openshift.io/v1/watch/securitycontextconstraints**
  - **GET**: watch individual changes to a list of SecurityContextConstraints. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/security.openshift.io/v1/securitycontextconstraints/{name}**
  - **DELETE**: delete SecurityContextConstraints
  - **GET**: read the specified SecurityContextConstraints
  - **PATCH**: partially update the specified SecurityContextConstraints
  - **PUT**: replace the specified SecurityContextConstraints
- **/apis/security.openshift.io/v1/watch/securitycontextconstraints/{name}**
  - **GET**: watch changes to an object of kind SecurityContextConstraints. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

9.2.1. /apis/security.openshift.io/v1/securitycontextconstraints

HTTP method

DELETE

Description

delete collection of SecurityContextConstraints

Table 9.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind SecurityContextConstraints

Table 9.2. HTTP responses

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

HTTP method

POST

Description

create SecurityContextConstraints

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	**SecurityContextConstraints** schema	

Table 9.5. HTTP responses

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

9.2.2. /apis/security.openshift.io/v1/watch/securitycontextconstraints

HTTP method

GET

Description

watch individual changes to a list of SecurityContextConstraints. deprecated: use the 'watch' parameter with a list operation instead.

Table 9.6. HTTP responses

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

9.2.3. /apis/security.openshift.io/v1/securitycontextconstraints/{name}

Table 9.7. Global path parameters

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

HTTP method

DELETE

Description

delete SecurityContextConstraints

Table 9.8. 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

Table 9.9. HTTP responses

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

HTTP method

GET

Description

read the specified SecurityContextConstraints

Table 9.10. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified SecurityContextConstraints

Table 9.11. Query parameters

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

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 9.12. HTTP responses

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

HTTP method

PUT

Description

replace the specified SecurityContextConstraints

Table 9.13. Query parameters

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

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 9.14. Body parameters

Parameter	Type	Description
body	**SecurityContextConstraints** schema	

Table 9.15. HTTP responses

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

9.2.4. /apis/security.openshift.io/v1/watch/securitycontextconstraints/{name}

Table 9.16. Global path parameters

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

HTTP method

GET

Description

watch changes to an object of kind SecurityContextConstraints. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 9.17. HTTP responses

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

CHAPTER 10. SERVICEACCOUNT [V1]

Description

ServiceAccount binds together: * a name, understood by users, and perhaps by peripheral systems, for an identity * a principal that can be authenticated and authorized * a set of secrets

Type

object

10.1. SPECIFICATION

Property	Type	Description
apiVersion	**string**	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
automountServiceAccountToken	**boolean**	AutomountServiceAccountToken indicates whether pods running as this service account should have an API token automatically mounted. Can be overridden at the pod level.
imagePullSecrets	**array**	ImagePullSecrets is a list of references to secrets in the same namespace to use for pulling any images in pods that reference this ServiceAccount. ImagePullSecrets are distinct from Secrets because Secrets can be mounted in the pod, but ImagePullSecrets are only accessed by the kubelet. More info: https://kubernetes.io/docs/concepts/containers/images/#specifying-imagepullsecrets-on-a-pod
imagePullSecrets[]	**object**	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Property	Type	Description
kind	**string**	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
secrets	**array**	Secrets is a list of the secrets in the same namespace that pods running using this ServiceAccount are allowed to use. Pods are only limited to this list if this service account has a "kubernetes.io/enforce-mountable-secrets" annotation set to "true". This field should not be used to find auto-generated service account token secrets for use outside of pods. Instead, tokens can be requested directly using the TokenRequest API, or service account token secrets can be manually created. More info: https://kubernetes.io/docs/concepts/configuration/secret
secrets[]	**object**	ObjectReference contains enough information to let you inspect or modify the referred object.

10.1.1. .imagePullSecrets

Description

ImagePullSecrets is a list of references to secrets in the same namespace to use for pulling any images in pods that reference this ServiceAccount. ImagePullSecrets are distinct from Secrets because Secrets can be mounted in the pod, but ImagePullSecrets are only accessed by the kubelet. More info: https://kubernetes.io/docs/concepts/containers/images/#specifying-imagepullsecrets-on-a-pod

Type

array

10.1.2. .imagePullSecrets[]

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

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

10.1.3. .secrets

Description

Secrets is a list of the secrets in the same namespace that pods running using this ServiceAccount are allowed to use. Pods are only limited to this list if this service account has a "kubernetes.io/enforce-mountable-secrets" annotation set to "true". This field should not be used to find auto-generated service account token secrets for use outside of pods. Instead, tokens can be requested directly using the TokenRequest API, or service account token secrets can be manually created. More info: https://kubernetes.io/docs/concepts/configuration/secret

Type

array

10.1.4. .secrets[]

Description

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

Type

object

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

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

10.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/serviceaccounts**
  - **GET**: list or watch objects of kind ServiceAccount
- **/api/v1/watch/serviceaccounts**
  - **GET**: watch individual changes to a list of ServiceAccount. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/serviceaccounts**
  - **DELETE**: delete collection of ServiceAccount
  - **GET**: list or watch objects of kind ServiceAccount
  - **POST**: create a ServiceAccount
- **/api/v1/watch/namespaces/{namespace}/serviceaccounts**
  - **GET**: watch individual changes to a list of ServiceAccount. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/serviceaccounts/{name}**
  - **DELETE**: delete a ServiceAccount
  - **GET**: read the specified ServiceAccount
  - **PATCH**: partially update the specified ServiceAccount
  - **PUT**: replace the specified ServiceAccount
- **/api/v1/watch/namespaces/{namespace}/serviceaccounts/{name}**
  - **GET**: watch changes to an object of kind ServiceAccount. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

10.2.1. /api/v1/serviceaccounts

HTTP method

GET

Description

list or watch objects of kind ServiceAccount

Table 10.1. HTTP responses

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

HTTP code	Reponse body
401 - Unauthorized	Empty

10.2.2. /api/v1/watch/serviceaccounts

HTTP method

GET

Description

watch individual changes to a list of ServiceAccount. deprecated: use the 'watch' parameter with a list operation instead.

Table 10.2. HTTP responses

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

10.2.3. /api/v1/namespaces/{namespace}/serviceaccounts

HTTP method

DELETE

Description

delete collection of ServiceAccount

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

Table 10.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind ServiceAccount

Table 10.5. HTTP responses

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

HTTP method

POST

Description

create a ServiceAccount

Table 10.6. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 10.7. Body parameters

Parameter	Type	Description
body	**ServiceAccount** schema	

Table 10.8. HTTP responses

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

10.2.4. /api/v1/watch/namespaces/{namespace}/serviceaccounts

HTTP method

GET

Description

watch individual changes to a list of ServiceAccount. deprecated: use the 'watch' parameter with a list operation instead.

Table 10.9. HTTP responses

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

10.2.5. /api/v1/namespaces/{namespace}/serviceaccounts/{name}

Table 10.10. Global path parameters

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

HTTP method

DELETE

Description

delete a ServiceAccount

Table 10.11. Query parameters

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

Table 10.12. HTTP responses

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

HTTP method

GET

Description

read the specified ServiceAccount

Table 10.13. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ServiceAccount

Table 10.14. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 10.15. HTTP responses

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

HTTP method

PUT

Description

replace the specified ServiceAccount

Table 10.16. Query parameters

Parameter	Type	Description
dryRun	**string**	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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 10.17. Body parameters

Parameter	Type	Description
body	**ServiceAccount** schema	

Table 10.18. HTTP responses

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

10.2.6. /api/v1/watch/namespaces/{namespace}/serviceaccounts/{name}

Table 10.19. Global path parameters

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

HTTP method

GET

Description

watch changes to an object of kind ServiceAccount. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 10.20. HTTP responses

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