OpenShift Container Platform 4.18

Policy APIs

Reference guide for policy APIs

Last Updated: 2025-05-15

OpenShift Container Platform 4.18 Policy APIs

Reference guide for policy APIs

Legal Notice

Copyright © 2025 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

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

Table of Contents

CHAPTER 1. POLICY APIS

1.1. EVICTION [POLICY/V1]

Description

Eviction evicts a pod from its node subject to certain policies and safety constraints. This is a subresource of Pod. A request to cause such an eviction is created by POSTing to …/pods/<pod name>/evictions.

Type

object

1.2. PODDISRUPTIONBUDGET [POLICY/V1]

Description

PodDisruptionBudget is an object to define the max disruption that can be caused to a collection of pods

Type

object

CHAPTER 2. EVICTION [POLICY/V1]

Description

Eviction evicts a pod from its node subject to certain policies and safety constraints. This is a subresource of Pod. A request to cause such an eviction is created by POSTing to …/pods/<pod name>/evictions.

Type

object

2.1. SPECIFICATION

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

2.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/namespaces/{namespace}/pods/{name}/eviction**
  - **POST**: create eviction of a Pod

2.2.1. /api/v1/namespaces/{namespace}/pods/{name}/eviction

Table 2.1. Global path parameters

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

Table 2.2. 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 eviction of a Pod

Table 2.3. Body parameters

Parameter	Type	Description
body	**Eviction** schema	

Table 2.4. HTTP responses

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

CHAPTER 3. PODDISRUPTIONBUDGET [POLICY/V1]

Description

PodDisruptionBudget is an object to define the max disruption that can be caused to a collection of pods

Type

object

3.1. SPECIFICATION

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

3.1.1. .spec

Description

PodDisruptionBudgetSpec is a description of a PodDisruptionBudget.

Type

object

Property	Type	Description
maxUnavailable	IntOrString	An eviction is allowed if at most "maxUnavailable" pods selected by "selector" are unavailable after the eviction, i.e. even in absence of the evicted pod. For example, one can prevent all voluntary evictions by specifying 0. This is a mutually exclusive setting with "minAvailable".
minAvailable	IntOrString	An eviction is allowed if at least "minAvailable" pods selected by "selector" will still be available after the eviction, i.e. even in the absence of the evicted pod. So for example you can prevent all voluntary evictions by specifying "100%".
selector	LabelSelector	Label query over pods whose evictions are managed by the disruption budget. A null selector will match no pods, while an empty ({}) selector will select all pods within the namespace.
unhealthyPodEvictionPolicy	**string**	UnhealthyPodEvictionPolicy defines the criteria for when unhealthy pods should be considered for eviction. Current implementation considers healthy pods, as pods that have status.conditions item with type="Ready",status="True". Valid policies are IfHealthyBudget and AlwaysAllow. If no policy is specified, the default behavior will be used, which corresponds to the IfHealthyBudget policy. IfHealthyBudget policy means that running pods (status.phase="Running"), but not

Property	Type	Description
		yet healthy can be evicted only if the guarded application is not disrupted (status.currentHealthy is at least equal to status.desiredHealthy). Healthy pods will be subject to the PDB for eviction. AlwaysAllow policy means that all running pods (status.phase="Running"), but not yet healthy are considered disrupted and can be evicted regardless of whether the criteria in a PDB is met. This means perspective running pods of a disrupted application might not get a chance to become healthy. Healthy pods will be subject to the PDB for eviction. Additional policies may be added in the future. Clients making eviction decisions should disallow eviction of unhealthy pods if they encounter an unrecognized policy in this field. This field is beta-level. The eviction API uses this field when the feature gate PDBUnhealthyPodEvictionPolicy is enabled (enabled by default). Possible enum values: - **"AlwaysAllow"** policy means that all running pods (status.phase="Running"), but not yet healthy are considered disrupted and can be evicted regardless of whether the criteria in a PDB is met. This means perspective running pods of a disrupted application might not get a chance to become healthy. Healthy pods will be subject to the PDB for eviction. - **"IfHealthyBudget"** policy means that running pods (status.phase="Running"), but not yet healthy can be evicted only if the guarded application is not disrupted (status.currentHealthy is at least equal to status.desiredHealthy). Healthy pods will be subject to the PDB for eviction.

Property	Type	Description

3.1.2. .status

Description

PodDisruptionBudgetStatus represents information about the status of a PodDisruptionBudget. Status may trail the actual state of a system.

Type

object

Required

- **disruptionsAllowed**
- **currentHealthy**
- **desiredHealthy**
- **expectedPods**

Property	Type	Description
conditions	**array (Condition)**	Conditions contain conditions for PDB. The disruption controller sets the DisruptionAllowed condition. The following are known values for the reason field (additional reasons could be added in the future): - SyncFailed: The controller encountered an error and wasn't able to compute the number of allowed disruptions. Therefore no disruptions are allowed and the status of the condition will be False. - InsufficientPods: The number of pods are either at or below the number required by the PodDisruptionBudget. No disruptions are allowed and the status of the condition will be False. - SufficientPods: There are more pods than required by the PodDisruptionBudget. The condition will be True, and the number of allowed disruptions are provided by the disruptionsAllowed property.
currentHealthy	**integer**	current number of healthy pods
desiredHealthy	**integer**	minimum desired number of healthy pods

Property	Type	Description
disruptedPods	**object (Time)**	DisruptedPods contains information about pods whose eviction was processed by the API server eviction subresource handler but has not yet been observed by the PodDisruptionBudget controller. A pod will be in this map from the time when the API server processed the eviction request to the time when the pod is seen by PDB controller as having been marked for deletion (or after a timeout). The key in the map is the name of the pod and the value is the time when the API server processed the eviction request. If the deletion didn't occur and a pod is still there it will be removed from the list automatically by PodDisruptionBudget controller after some time. If everything goes smooth this map should be empty for the most of the time. Large number of entries in the map may indicate problems with pod deletions.
disruptionsAllowed	**integer**	Number of pod disruptions that are currently allowed.
expectedPods	**integer**	total number of pods counted by this disruption budget
observedGeneration	**integer**	Most recent generation observed when updating this PDB status. DisruptionsAllowed and other status information is valid only if observedGeneration equals to PDB's object generation.

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/policy/v1/poddisruptionbudgets**
  - **GET**: list or watch objects of kind PodDisruptionBudget
- **/apis/policy/v1/watch/poddisruptionbudgets**

- **GET**: watch individual changes to a list of PodDisruptionBudget. deprecated: use the 'watch' parameter with a list operation instead.

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

3.2.1. /apis/policy/v1/poddisruptionbudgets

HTTP method

GET

Description

list or watch objects of kind PodDisruptionBudget

Table 3.1. HTTP responses

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

3.2.2. /apis/policy/v1/watch/poddisruptionbudgets

HTTP method

GET

Description

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

Table 3.2. HTTP responses

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

3.2.3. /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets

HTTP method

DELETE

Description

delete collection of PodDisruptionBudget

Table 3.3. Query parameters

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

Table 3.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind PodDisruptionBudget

Table 3.5. HTTP responses

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

HTTP method

POST

Description

create a PodDisruptionBudget

Table 3.6. 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 3.7. Body parameters

Parameter	Type	Description
body	**PodDisruptionBudget** schema	

Table 3.8. HTTP responses

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

3.2.4. /apis/policy/v1/watch/namespaces/{namespace}/poddisruptionbudgets

HTTP method

GET

Description

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

Table 3.9. HTTP responses

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

3.2.5. /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}

Table 3.10. Global path parameters

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

HTTP method

DELETE

Description

delete a PodDisruptionBudget

Table 3.11. 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 3.12. HTTP responses

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

HTTP method

GET

Description

read the specified PodDisruptionBudget

Table 3.13. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified PodDisruptionBudget

Table 3.14. 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 3.15. HTTP responses

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

HTTP method

PUT

Description

replace the specified PodDisruptionBudget

Table 3.16. 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 3.17. Body parameters

Parameter	Type	Description
body	**PodDisruptionBudget** schema	

Table 3.18. HTTP responses

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

3.2.6. /apis/policy/v1/watch/namespaces/{namespace}/poddisruptionbudgets/{nam

Table 3.19. Global path parameters

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

HTTP method

GET

Description

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

Table 3.20. HTTP responses

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

3.2.7. /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}/stat

Table 3.21. Global path parameters

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

HTTP method

GET

Description

read status of the specified PodDisruptionBudget

Table 3.22. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified PodDisruptionBudget

Table 3.23. Query parameters

Parameter	Type	Description

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

Table 3.24. HTTP responses

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

HTTP method

PUT

Description

replace status of the specified PodDisruptionBudget

Table 3.25. Query parameters

Parameter	Type	Description

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

Table 3.26. Body parameters

Parameter	Type	Description
body	**PodDisruptionBudget** schema	

Table 3.27. HTTP responses

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