



# OpenShift Container Platform 4.18

## Network APIs

Reference guide for network APIs



# OpenShift Container Platform 4.18 Network APIs

---

Reference guide for network 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js<sup>®</sup> 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<sup>®</sup> 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 network API objects and their detailed specifications.

## Table of Contents

<b>CHAPTER 1. NETWORK APIS</b> .....	<b>12</b>
1.1. ADMINNETWORKPOLICY [POLICY.NETWORKING.K8S.IO/V1ALPHA1]	12
1.2. ADMINPOLICYBASEDEXTERNALROUTE [K8S.OVN.ORG/V1]	12
1.3. BASELINEADMINNETWORKPOLICY [POLICY.NETWORKING.K8S.IO/V1ALPHA1]	12
1.4. CLOUDPRIVATEIPCONFIG [CLOUD.NETWORK.OPENSIFT.IO/V1]	12
1.5. EGRESSFIREWALL [K8S.OVN.ORG/V1]	12
1.6. EGRESSIP [K8S.OVN.ORG/V1]	13
1.7. EGRESSQOS [K8S.OVN.ORG/V1]	13
1.8. EGRESSSERVICE [K8S.OVN.ORG/V1]	13
1.9. ENDPOINTS [V1]	13
1.10. ENDPOINTSLSICE [DISCOVERY.K8S.IO/V1]	14
1.11. EGRESSROUTER [NETWORK.OPERATOR.OPENSIFT.IO/V1]	14
1.12. INGRESS [NETWORKING.K8S.IO/V1]	14
1.13. INGRESSCLASS [NETWORKING.K8S.IO/V1]	14
1.14. IPPOOL [WHEREABOUTS.CNI.CNCF.IO/V1ALPHA1]	15
1.15. MULTINETWORKPOLICY [K8S.CNI.CNCF.IO/V1BETA1]	15
1.16. NETWORKATTACHMENTDEFINITION [K8S.CNI.CNCF.IO/V1]	15
1.17. NETWORKPOLICY [NETWORKING.K8S.IO/V1]	15
1.18. OVERLAPPINGRANGEIPRESERVATION [WHEREABOUTS.CNI.CNCF.IO/V1ALPHA1]	15
1.19. PODNETWORKCONNECTIVITYCHECK [CONTROLPLANE.OPERATOR.OPENSIFT.IO/V1ALPHA1]	15
1.20. ROUTE [ROUTE.OPENSIFT.IO/V1]	16
1.21. SERVICE [V1]	16
<b>CHAPTER 2. ADMINNETWORKPOLICY [POLICY.NETWORKING.K8S.IO/V1ALPHA1]</b> .....	<b>17</b>
2.1. SPECIFICATION	17
2.1.1. .spec	18
2.1.2. .spec.egress	19
2.1.3. .spec.egress[]	20
2.1.4. .spec.egress[].ports	21
2.1.5. .spec.egress[].ports[]	22
2.1.6. .spec.egress[].ports[].portNumber	22
2.1.7. .spec.egress[].ports[].portRange	23
2.1.8. .spec.egress[].to	24
2.1.9. .spec.egress[].to[]	24
2.1.10. .spec.egress[].to[].namespaces	25
2.1.11. .spec.egress[].to[].namespaces.matchExpressions	26
2.1.12. .spec.egress[].to[].namespaces.matchExpressions[]	26
2.1.13. .spec.egress[].to[].nodes	27
2.1.14. .spec.egress[].to[].nodes.matchExpressions	28
2.1.15. .spec.egress[].to[].nodes.matchExpressions[]	28
2.1.16. .spec.egress[].to[].pods	29
2.1.17. .spec.egress[].to[].pods.namespaceSelector	29
2.1.18. .spec.egress[].to[].pods.namespaceSelector.matchExpressions	30
2.1.19. .spec.egress[].to[].pods.namespaceSelector.matchExpressions[]	30
2.1.20. .spec.egress[].to[].pods.podSelector	31
2.1.21. .spec.egress[].to[].pods.podSelector.matchExpressions	31
2.1.22. .spec.egress[].to[].pods.podSelector.matchExpressions[]	32
2.1.23. .spec.ingress	32
2.1.24. .spec.ingress[]	32
2.1.25. .spec.ingress[].from	34
2.1.26. .spec.ingress[].from[]	35

2.1.27. .spec.ingress[].from[].namespaces	35
2.1.28. .spec.ingress[].from[].namespaces.matchExpressions	36
2.1.29. .spec.ingress[].from[].namespaces.matchExpressions[]	36
2.1.30. .spec.ingress[].from[].pods	37
2.1.31. .spec.ingress[].from[].pods.namespaceSelector	37
2.1.32. .spec.ingress[].from[].pods.namespaceSelector.matchExpressions	38
2.1.33. .spec.ingress[].from[].pods.namespaceSelector.matchExpressions[]	38
2.1.34. .spec.ingress[].from[].pods.podSelector	39
2.1.35. .spec.ingress[].from[].pods.podSelector.matchExpressions	39
2.1.36. .spec.ingress[].from[].pods.podSelector.matchExpressions[]	40
2.1.37. .spec.ingress[].ports	40
2.1.38. .spec.ingress[].ports[]	40
2.1.39. .spec.ingress[].ports[].portNumber	41
2.1.40. .spec.ingress[].ports[].portRange	42
2.1.41. .spec.subject	42
2.1.42. .spec.subject.namespaces	43
2.1.43. .spec.subject.namespaces.matchExpressions	43
2.1.44. .spec.subject.namespaces.matchExpressions[]	44
2.1.45. .spec.subject.pods	44
2.1.46. .spec.subject.pods.namespaceSelector	45
2.1.47. .spec.subject.pods.namespaceSelector.matchExpressions	45
2.1.48. .spec.subject.pods.namespaceSelector.matchExpressions[]	45
2.1.49. .spec.subject.pods.podSelector	46
2.1.50. .spec.subject.pods.podSelector.matchExpressions	47
2.1.51. .spec.subject.pods.podSelector.matchExpressions[]	47
2.1.52. .status	48
2.1.53. .status.conditions	49
2.1.54. .status.conditions[]	49
2.2. API ENDPOINTS	51
2.2.1. /apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies	52
2.2.2. /apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies/{name}	53
2.2.3. /apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies/{name}/status	56
<b>CHAPTER 3. ADMINPOLICYBASEDEXTERNALROUTE [K8S.OVN.ORG/V1]</b>	<b>60</b>
3.1. SPECIFICATION	60
3.1.1. .spec	61
3.1.2. .spec.from	61
3.1.3. .spec.from.namespaceSelector	62
3.1.4. .spec.from.namespaceSelector.matchExpressions	62
3.1.5. .spec.from.namespaceSelector.matchExpressions[]	62
3.1.6. .spec.nextHops	63
3.1.7. .spec.nextHops.dynamic	64
3.1.8. .spec.nextHops.dynamic[]	64
3.1.9. .spec.nextHops.dynamic[].namespaceSelector	65
3.1.10. .spec.nextHops.dynamic[].namespaceSelector.matchExpressions	66
3.1.11. .spec.nextHops.dynamic[].namespaceSelector.matchExpressions[]	66
3.1.12. .spec.nextHops.dynamic[].podSelector	67
3.1.13. .spec.nextHops.dynamic[].podSelector.matchExpressions	67
3.1.14. .spec.nextHops.dynamic[].podSelector.matchExpressions[]	68
3.1.15. .spec.nextHops.static	68
3.1.16. .spec.nextHops.static[]	68
3.1.17. .status	69
3.2. API ENDPOINTS	69

3.2.1. /apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes	70
3.2.2. /apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes/{name}	72
3.2.3. /apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes/{name}/status	74
<b>CHAPTER 4. BASELINEADMINNETWORKPOLICY [POLICY.NETWORKING.K8S.IO/V1ALPHA1] .....</b>	<b>78</b>
4.1. SPECIFICATION	78
4.1.1. .spec	79
4.1.2. .spec.egress	80
4.1.3. .spec.egress[]	80
4.1.4. .spec.egress[].ports	82
4.1.5. .spec.egress[].ports[]	82
4.1.6. .spec.egress[].ports[].portNumber	83
4.1.7. .spec.egress[].ports[].portRange	83
4.1.8. .spec.egress[].to	84
4.1.9. .spec.egress[].to[]	84
4.1.10. .spec.egress[].to[].namespaces	86
4.1.11. .spec.egress[].to[].namespaces.matchExpressions	86
4.1.12. .spec.egress[].to[].namespaces.matchExpressions[]	86
4.1.13. .spec.egress[].to[].nodes	87
4.1.14. .spec.egress[].to[].nodes.matchExpressions	88
4.1.15. .spec.egress[].to[].nodes.matchExpressions[]	88
4.1.16. .spec.egress[].to[].pods	89
4.1.17. .spec.egress[].to[].pods.namespaceSelector	89
4.1.18. .spec.egress[].to[].pods.namespaceSelector.matchExpressions	90
4.1.19. .spec.egress[].to[].pods.namespaceSelector.matchExpressions[]	90
4.1.20. .spec.egress[].to[].pods.podSelector	91
4.1.21. .spec.egress[].to[].pods.podSelector.matchExpressions	91
4.1.22. .spec.egress[].to[].pods.podSelector.matchExpressions[]	92
4.1.23. .spec.ingress	92
4.1.24. .spec.ingress[]	93
4.1.25. .spec.ingress[].from	94
4.1.26. .spec.ingress[].from[]	94
4.1.27. .spec.ingress[].from[].namespaces	95
4.1.28. .spec.ingress[].from[].namespaces.matchExpressions	96
4.1.29. .spec.ingress[].from[].namespaces.matchExpressions[]	96
4.1.30. .spec.ingress[].from[].pods	96
4.1.31. .spec.ingress[].from[].pods.namespaceSelector	97
4.1.32. .spec.ingress[].from[].pods.namespaceSelector.matchExpressions	97
4.1.33. .spec.ingress[].from[].pods.namespaceSelector.matchExpressions[]	98
4.1.34. .spec.ingress[].from[].pods.podSelector	98
4.1.35. .spec.ingress[].from[].pods.podSelector.matchExpressions	99
4.1.36. .spec.ingress[].from[].pods.podSelector.matchExpressions[]	99
4.1.37. .spec.ingress[].ports	100
4.1.38. .spec.ingress[].ports[]	100
4.1.39. .spec.ingress[].ports[].portNumber	101
4.1.40. .spec.ingress[].ports[].portRange	101
4.1.41. .spec.subject	102
4.1.42. .spec.subject.namespaces	102
4.1.43. .spec.subject.namespaces.matchExpressions	103
4.1.44. .spec.subject.namespaces.matchExpressions[]	103
4.1.45. .spec.subject.pods	104
4.1.46. .spec.subject.pods.namespaceSelector	104
4.1.47. .spec.subject.pods.namespaceSelector.matchExpressions	105

4.1.48. .spec.subject.pods.namespaceSelector.matchExpressions[]	105
4.1.49. .spec.subject.pods.podSelector	106
4.1.50. .spec.subject.pods.podSelector.matchExpressions	106
4.1.51. .spec.subject.pods.podSelector.matchExpressions[]	107
4.1.52. .status	107
4.1.53. .status.conditions	108
4.1.54. .status.conditions[]	108
4.2. API ENDPOINTS	110
4.2.1. /apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies	110
4.2.2. /apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies/{name}	112
4.2.3. /apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies/{name}/status	115
<b>CHAPTER 5. CLOUDPRIVATEIPCONFIG [CLOUD.NETWORK.OPENSIFT.IO/V1]</b>	<b>119</b>
5.1. SPECIFICATION	119
5.1.1. .spec	120
5.1.2. .status	120
5.1.3. .status.conditions	121
5.1.4. .status.conditions[]	121
5.2. API ENDPOINTS	122
5.2.1. /apis/cloud.network.openshift.io/v1/cloudprivateipconfigs	123
5.2.2. /apis/cloud.network.openshift.io/v1/cloudprivateipconfigs/{name}	125
5.2.3. /apis/cloud.network.openshift.io/v1/cloudprivateipconfigs/{name}/status	127
<b>CHAPTER 6. EGRESSFIREWALL [K8S.OVN.ORG/V1]</b>	<b>131</b>
6.1. SPECIFICATION	131
6.1.1. .spec	132
6.1.2. .spec.egress	132
6.1.3. .spec.egress[]	132
6.1.4. .spec.egress[].ports	133
6.1.5. .spec.egress[].ports[]	133
6.1.6. .spec.egress[].to	133
6.1.7. .spec.egress[].to.nodeSelector	134
6.1.8. .spec.egress[].to.nodeSelector.matchExpressions	135
6.1.9. .spec.egress[].to.nodeSelector.matchExpressions[]	135
6.1.10. .status	136
6.2. API ENDPOINTS	136
6.2.1. /apis/k8s.ovn.org/v1/egressfirewalls	137
6.2.2. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls	137
6.2.3. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls/{name}	139
6.2.4. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls/{name}/status	142
<b>CHAPTER 7. EGRESSIP [K8S.OVN.ORG/V1]</b>	<b>145</b>
7.1. SPECIFICATION	145
7.1.1. .spec	146
7.1.2. .spec.namespaceSelector	146
7.1.3. .spec.namespaceSelector.matchExpressions	147
7.1.4. .spec.namespaceSelector.matchExpressions[]	147
7.1.5. .spec.podSelector	148
7.1.6. .spec.podSelector.matchExpressions	148
7.1.7. .spec.podSelector.matchExpressions[]	149
7.1.8. .status	149
7.1.9. .status.items	150
7.1.10. .status.items[]	150
7.2. API ENDPOINTS	150



7.2.1. /apis/k8s.ovn.org/v1/egressips	151
7.2.2. /apis/k8s.ovn.org/v1/egressips/{name}	152
<b>CHAPTER 8. EGRESSQOS [K8S.OVN.ORG/V1]</b>	<b>156</b>
8.1. SPECIFICATION	156
8.1.1. .spec	157
8.1.2. .spec.egress	157
8.1.3. .spec.egress[]	157
8.1.4. .spec.egress[].podSelector	158
8.1.5. .spec.egress[].podSelector.matchExpressions	158
8.1.6. .spec.egress[].podSelector.matchExpressions[]	159
8.1.7. .status	159
8.1.8. .status.conditions	160
8.1.9. .status.conditions[]	160
8.2. API ENDPOINTS	162
8.2.1. /apis/k8s.ovn.org/v1/egressqoses	163
8.2.2. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses	163
8.2.3. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses/{name}	165
8.2.4. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses/{name}/status	168
<b>CHAPTER 9. EGRESSSERVICE [K8S.OVN.ORG/V1]</b>	<b>171</b>
9.1. SPECIFICATION	171
9.1.1. .spec	172
9.1.2. .spec.nodeSelector	172
9.1.3. .spec.nodeSelector.matchExpressions	173
9.1.4. .spec.nodeSelector.matchExpressions[]	173
9.1.5. .status	174
9.2. API ENDPOINTS	174
9.2.1. /apis/k8s.ovn.org/v1/egressservices	175
9.2.2. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices	175
9.2.3. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices/{name}	177
9.2.4. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices/{name}/status	180
<b>CHAPTER 10. ENDPOINTS [V1]</b>	<b>184</b>
10.1. SPECIFICATION	184
10.1.1. .subsets	185
10.1.2. .subsets[]	186
10.1.3. .subsets[].addresses	186
10.1.4. .subsets[].addresses[]	187
10.1.5. .subsets[].addresses[].targetRef	187
10.1.6. .subsets[].notReadyAddresses	189
10.1.7. .subsets[].notReadyAddresses[]	189
10.1.8. .subsets[].notReadyAddresses[].targetRef	189
10.1.9. .subsets[].ports	191
10.1.10. .subsets[].ports[]	191
10.2. API ENDPOINTS	193
10.2.1. /api/v1/endpoints	193
10.2.2. /api/v1/watch/endpoints	194
10.2.3. /api/v1/namespaces/{namespace}/endpoints	194
10.2.4. /api/v1/watch/namespaces/{namespace}/endpoints	196
10.2.5. /api/v1/namespaces/{namespace}/endpoints/{name}	197
10.2.6. /api/v1/watch/namespaces/{namespace}/endpoints/{name}	200
<b>CHAPTER 11. ENDPOINTSLSICE [DISCOVERY.K8S.IO/V1]</b>	<b>201</b>

11.1. SPECIFICATION	201
11.1.1. .endpoints	202
11.1.2. .endpoints[]	202
11.1.3. .endpoints[].conditions	204
11.1.4. .endpoints[].hints	205
11.1.5. .endpoints[].hints.forZones	205
11.1.6. .endpoints[].hints.forZones[]	205
11.1.7. .ports	206
11.1.8. .ports[]	206
11.2. API ENDPOINTS	208
11.2.1. /apis/discovery.k8s.io/v1/endpointslices	209
11.2.2. /apis/discovery.k8s.io/v1/watch/endpointslices	209
11.2.3. /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices	210
11.2.4. /apis/discovery.k8s.io/v1/watch/namespaces/{namespace}/endpointslices	211
11.2.5. /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices/{name}	212
11.2.6. /apis/discovery.k8s.io/v1/watch/namespaces/{namespace}/endpointslices/{name}	215
<b>CHAPTER 12. EGRESSROUTER [NETWORK.OPERATOR.OPENSIFT.IO/V1]</b>	<b>216</b>
12.1. SPECIFICATION	216
12.1.1. .spec	217
12.1.2. .spec.addresses	218
12.1.3. .spec.addresses[]	218
12.1.4. .spec.networkInterface	218
12.1.5. .spec.networkInterface.macvlan	219
12.1.6. .spec.redirect	219
12.1.7. .spec.redirect.redirectRules	220
12.1.8. .spec.redirect.redirectRules[]	220
12.1.9. .status	221
12.1.10. .status.conditions	221
12.1.11. .status.conditions[]	221
12.2. API ENDPOINTS	222
12.2.1. /apis/network.operator.openshift.io/v1/egressrouters	223
12.2.2. /apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrouters	223
12.2.3. /apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrouters/{name}	225
12.2.4. /apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrouters/{name}/status	228
<b>CHAPTER 13. INGRESS [NETWORKING.K8S.IO/V1]</b>	<b>232</b>
13.1. SPECIFICATION	232
13.1.1. .spec	232
13.1.2. .spec.defaultBackend	234
13.1.3. .spec.defaultBackend.service	234
13.1.4. .spec.defaultBackend.service.port	235
13.1.5. .spec.rules	235
13.1.6. .spec.rules[]	235
13.1.7. .spec.rules[].http	237
13.1.8. .spec.rules[].http.paths	237
13.1.9. .spec.rules[].http.paths[]	237
13.1.10. .spec.rules[].http.paths[].backend	239
13.1.11. .spec.rules[].http.paths[].backend.service	240
13.1.12. .spec.rules[].http.paths[].backend.service.port	240
13.1.13. .spec.tls	241
13.1.14. .spec.tls[]	241
13.1.15. .status	242

13.1.16. .status.loadBalancer	242
13.1.17. .status.loadBalancer.ingress	242
13.1.18. .status.loadBalancer.ingress[]	242
13.1.19. .status.loadBalancer.ingress[].ports	243
13.1.20. .status.loadBalancer.ingress[].ports[]	243
13.2. API ENDPOINTS	244
13.2.1. /apis/networking.k8s.io/v1/ingresses	245
13.2.2. /apis/networking.k8s.io/v1/watch/ingresses	245
13.2.3. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses	246
13.2.4. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses	247
13.2.5. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}	248
13.2.6. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses/{name}	251
13.2.7. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status	251
<b>CHAPTER 14. INGRESSCLASS [NETWORKING.K8S.IO/V1]</b>	<b>255</b>
14.1. SPECIFICATION	255
14.1.1. .spec	255
14.1.2. .spec.parameters	256
14.2. API ENDPOINTS	257
14.2.1. /apis/networking.k8s.io/v1/ingressclasses	257
14.2.2. /apis/networking.k8s.io/v1/watch/ingressclasses	259
14.2.3. /apis/networking.k8s.io/v1/ingressclasses/{name}	260
14.2.4. /apis/networking.k8s.io/v1/watch/ingressclasses/{name}	263
<b>CHAPTER 15. IPPOL [WHEREABOUTS.CNI.CNCF.IO/V1ALPHA1]</b>	<b>264</b>
15.1. SPECIFICATION	264
15.1.1. .spec	264
15.1.2. .spec.allocations	265
15.1.3. .spec.allocations{}	265
15.2. API ENDPOINTS	266
15.2.1. /apis/whereabouts.cni.cncf.io/v1alpha1/ippools	266
15.2.2. /apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/ippools	266
15.2.3. /apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/ippools/{name}	268
<b>CHAPTER 16. MULTINETWORKPOLICY [K8S.CNI.CNCF.IO/V1BETA1]</b>	<b>272</b>
16.1. SPECIFICATION	272
16.1.1. .spec	272
16.1.2. .spec.egress	274
16.1.3. .spec.egress[]	275
16.1.4. .spec.egress[].ports	276
16.1.5. .spec.egress[].ports[]	276
16.1.6. .spec.egress[].to	276
16.1.7. .spec.egress[].to[]	276
16.1.8. .spec.egress[].to[].ipBlock	277
16.1.9. .spec.egress[].to[].namespaceSelector	278
16.1.10. .spec.egress[].to[].namespaceSelector.matchExpressions	278
16.1.11. .spec.egress[].to[].namespaceSelector.matchExpressions[]	278
16.1.12. .spec.egress[].to[].podSelector	279
16.1.13. .spec.egress[].to[].podSelector.matchExpressions	280
16.1.14. .spec.egress[].to[].podSelector.matchExpressions[]	280
16.1.15. .spec.ingress	281
16.1.16. .spec.ingress[]	281
16.1.17. .spec.ingress[].from	282
16.1.18. .spec.ingress[].from[]	282

16.1.19. .spec.ingress[].from[].ipBlock	283
16.1.20. .spec.ingress[].from[].namespaceSelector	284
16.1.21. .spec.ingress[].from[].namespaceSelector.matchExpressions	284
16.1.22. .spec.ingress[].from[].namespaceSelector.matchExpressions[]	285
16.1.23. .spec.ingress[].from[].podSelector	285
16.1.24. .spec.ingress[].from[].podSelector.matchExpressions	286
16.1.25. .spec.ingress[].from[].podSelector.matchExpressions[]	286
16.1.26. .spec.ingress[].ports	287
16.1.27. .spec.ingress[].ports[]	287
16.1.28. .spec.podSelector	287
16.1.29. .spec.podSelector.matchExpressions	288
16.1.30. .spec.podSelector.matchExpressions[]	288
16.2. API ENDPOINTS	289
16.2.1. /apis/k8s.cni.cncf.io/v1beta1/multi-networkpolicies	289
16.2.2. /apis/k8s.cni.cncf.io/v1beta1/namespaces/{namespace}/multi-networkpolicies	290
16.2.3. /apis/k8s.cni.cncf.io/v1beta1/namespaces/{namespace}/multi-networkpolicies/{name}	291
<b>CHAPTER 17. NETWORKATTACHMENTDEFINITION [K8S.CNI.CNCF.IO/V1]</b>	<b>295</b>
17.1. SPECIFICATION	295
17.1.1. .spec	295
17.2. API ENDPOINTS	296
17.2.1. /apis/k8s.cni.cncf.io/v1/network-attachment-definitions	296
17.2.2. /apis/k8s.cni.cncf.io/v1/namespaces/{namespace}/network-attachment-definitions	297
17.2.3. /apis/k8s.cni.cncf.io/v1/namespaces/{namespace}/network-attachment-definitions/{name}	298
<b>CHAPTER 18. NETWORKPOLICY [NETWORKING.K8S.IO/V1]</b>	<b>302</b>
18.1. SPECIFICATION	302
18.1.1. .spec	302
18.1.2. .spec.egress	304
18.1.3. .spec.egress[]	305
18.1.4. .spec.egress[].ports	306
18.1.5. .spec.egress[].ports[]	306
18.1.6. .spec.egress[].to	306
18.1.7. .spec.egress[].to[]	307
18.1.8. .spec.egress[].to[].ipBlock	308
18.1.9. .spec.ingress	308
18.1.10. .spec.ingress[]	309
18.1.11. .spec.ingress[].from	310
18.1.12. .spec.ingress[].from[]	310
18.1.13. .spec.ingress[].from[].ipBlock	311
18.1.14. .spec.ingress[].ports	311
18.1.15. .spec.ingress[].ports[]	312
18.2. API ENDPOINTS	312
18.2.1. /apis/networking.k8s.io/v1/networkpolicies	313
18.2.2. /apis/networking.k8s.io/v1/watch/networkpolicies	313
18.2.3. /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies	314
18.2.4. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies	316
18.2.5. /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}	316
18.2.6. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies/{name}	319
<b>CHAPTER 19. OVERLAPPINGRANGEIPRESERVATION [WHEREABOUTS.CNI.CNCF.IO/V1ALPHA1]</b>	<b>321</b>
19.1. SPECIFICATION	321
19.1.1. .spec	322
19.2. API ENDPOINTS	322

19.2.1. /apis/whereabouts.cni.cncf.io/v1alpha1/overlappingrangeipreservations	322
19.2.2. /apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/overlappingrangeipreservations	323
19.2.3. /apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/overlappingrangeipreservations/{name}	324
 <b>CHAPTER 20. PODNETWORKCONNECTIVITYCHECK</b>	
<b>[CONTROLPLANE.OPERATOR.OPENSIFT.IO/V1ALPHA1]</b> .....	<b>328</b>
20.1. SPECIFICATION	328
20.1.1. .spec	329
20.1.2. .spec.tlsClientCert	329
20.1.3. .status	330
20.1.4. .status.conditions	331
20.1.5. .status.conditions[]	331
20.1.6. .status.failures	331
20.1.7. .status.failures[]	331
20.1.8. .status.outages	332
20.1.9. .status.outages[]	332
20.1.10. .status.outages[].endLogs	333
20.1.11. .status.outages[].endLogs[]	333
20.1.12. .status.outages[].startLogs	334
20.1.13. .status.outages[].startLogs[]	334
20.1.14. .status.successes	335
20.1.15. .status.successes[]	335
20.2. API ENDPOINTS	335
20.2.1. /apis/controlplane.operator.openshift.io/v1alpha1/podnetworkconnectivitychecks	336
20.2.2. /apis/controlplane.operator.openshift.io/v1alpha1/namespaces/{namespace}/podnetworkconnectivitychecks	336
20.2.3. /apis/controlplane.operator.openshift.io/v1alpha1/namespaces/{namespace}/podnetworkconnectivitychecks/{name}	338
20.2.4. /apis/controlplane.operator.openshift.io/v1alpha1/namespaces/{namespace}/podnetworkconnectivitychecks/{name}/status	341
 <b>CHAPTER 21. ROUTE [ROUTE.OPENSIFT.IO/V1]</b> .....	
<b>21.1. SPECIFICATION</b>	<b>345</b>
21.1.1. .spec	347
21.1.2. .spec.alternateBackends	349
21.1.3. .spec.alternateBackends[]	350
21.1.4. .spec.httpHeaders	350
21.1.5. .spec.httpHeaders.actions	351
21.1.6. .spec.httpHeaders.actions.request	353
21.1.7. .spec.httpHeaders.actions.request[]	354
21.1.8. .spec.httpHeaders.actions.request[].action	354
21.1.9. .spec.httpHeaders.actions.request[].action.set	355
21.1.10. .spec.httpHeaders.actions.response	356
21.1.11. .spec.httpHeaders.actions.response[]	356
21.1.12. .spec.httpHeaders.actions.response[].action	357
21.1.13. .spec.httpHeaders.actions.response[].action.set	357
21.1.14. .spec.port	358
21.1.15. .spec.tls	358
21.1.16. .spec.tls.externalCertificate	360
21.1.17. .spec.to	361
21.1.18. .status	361
21.1.19. .status.ingress	362

21.1.20. .status.ingress[]	362
21.1.21. .status.ingress[].conditions	363
21.1.22. .status.ingress[].conditions[]	363
21.2. API ENDPOINTS	364
21.2.1. /apis/route.openshift.io/v1/routes	365
21.2.2. /apis/route.openshift.io/v1/watch/routes	365
21.2.3. /apis/route.openshift.io/v1/namespaces/{namespace}/routes	365
21.2.4. /apis/route.openshift.io/v1/watch/namespaces/{namespace}/routes	367
21.2.5. /apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}	368
21.2.6. /apis/route.openshift.io/v1/watch/namespaces/{namespace}/routes/{name}	371
21.2.7. /apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}/status	371
<b>CHAPTER 22. SERVICE [V1]</b> .....	<b>375</b>
22.1. SPECIFICATION	375
22.1.1. .spec	376
22.1.2. .spec.ports	388
22.1.3. .spec.ports[]	388
22.1.4. .spec.sessionAffinityConfig	391
22.1.5. .spec.sessionAffinityConfig.clientIP	391
22.1.6. .status	392
22.1.7. .status.loadBalancer	392
22.1.8. .status.loadBalancer.ingress	392
22.1.9. .status.loadBalancer.ingress[]	393
22.1.10. .status.loadBalancer.ingress[].ports	393
22.1.11. .status.loadBalancer.ingress[].ports[]	394
22.2. API ENDPOINTS	394
22.2.1. /api/v1/services	395
22.2.2. /api/v1/watch/services	396
22.2.3. /api/v1/namespaces/{namespace}/services	396
22.2.4. /api/v1/watch/namespaces/{namespace}/services	398
22.2.5. /api/v1/namespaces/{namespace}/services/{name}	398
22.2.6. /api/v1/watch/namespaces/{namespace}/services/{name}	401
22.2.7. /api/v1/namespaces/{namespace}/services/{name}/status	402



## CHAPTER 1. NETWORK APIS

### 1.1. ADMINNETWORKPOLICY

[POLICY.NETWORKING.K8S.IO/V1ALPHA1]

#### Description

AdminNetworkPolicy is a cluster level resource that is part of the AdminNetworkPolicy API.

#### Type

**object**

### 1.2. ADMINPOLICYBASEDEXTERNALROUTE [K8S.OVN.ORG/V1]

#### Description

AdminPolicyBasedExternalRoute is a CRD allowing the cluster administrators to configure policies for external gateway IPs to be applied to all the pods contained in selected namespaces. Egress traffic from the pods that belong to the selected namespaces to outside the cluster is routed through these external gateway IPs.

#### Type

**object**

### 1.3. BASELINEADMINNETWORKPOLICY

[POLICY.NETWORKING.K8S.IO/V1ALPHA1]

#### Description

BaselineAdminNetworkPolicy is a cluster level resource that is part of the AdminNetworkPolicy API.

#### Type

**object**

### 1.4. CLOUDPRIVATEIPCONFIG [CLOUD.NETWORK.OPENSIFT.IO/V1]

#### Description

CloudPrivateIPConfig performs an assignment of a private IP address to the primary NIC associated with cloud VMs. This is done by specifying the IP and Kubernetes node which the IP should be assigned to. This CRD is intended to be used by the network plugin which manages the cluster network. The spec side represents the desired state requested by the network plugin, and the status side represents the current state that this CRD's controller has executed. No users will have permission to modify it, and if a cluster-admin decides to edit it for some reason, their changes will be overwritten the next time the network plugin reconciles the object. Note: the CR's name must specify the requested private IP address (can be IPv4 or IPv6).

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

#### Type

**object**

### 1.5. EGRESSFIREWALL [K8S.OVN.ORG/V1]

#### Description



EgressFirewall describes the current egress firewall for a Namespace. Traffic from a pod to an IP address outside the cluster will be checked against each EgressFirewallRule in the pod's namespace's EgressFirewall, in order. If no rule matches (or no EgressFirewall is present) then the traffic will be allowed by default.

#### Type

**object**

## 1.6. EGRESSIP [K8S.OVN.ORG/V1]

#### Description

EgressIP is a CRD allowing the user to define a fixed source IP for all egress traffic originating from any pods which match the EgressIP resource according to its spec definition.

#### Type

**object**

## 1.7. EGRESSQOS [K8S.OVN.ORG/V1]

#### Description

EgressQoS is a CRD that allows the user to define a DSCP value for pods egress traffic on its namespace to specified CIDRs. Traffic from these pods will be checked against each EgressQoSRule in the namespace's EgressQoS, and if there is a match the traffic is marked with the relevant DSCP value.

#### Type

**object**

## 1.8. EGRESSSERVICE [K8S.OVN.ORG/V1]

#### Description

EgressService is a CRD that allows the user to request that the source IP of egress packets originating from all of the pods that are endpoints of the corresponding LoadBalancer Service would be its ingress IP. In addition, it allows the user to request that egress packets originating from all of the pods that are endpoints of the LoadBalancer service would use a different network than the main one.

#### Type

**object**

## 1.9. ENDPOINTS [V1]

#### Description

Endpoints is a collection of endpoints that implement the actual service. Example:

```
Name: "mysvc",
Subsets: [
  {
    Addresses: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}],
    Ports: [{"name": "a", "port": 8675}, {"name": "b", "port": 309}]
  },
  {
    Addresses: [{"ip": "10.10.3.3"}],
```

```
Ports: [{"name": "a", "port": 93}, {"name": "b", "port": 76}]
},
]
```

Type

**object**

## 1.10. ENDPOINTSICE [DISCOVERY.K8S.IO/V1]

### Description

EndpointSlice represents a subset of the endpoints that implement a service. For a given service there may be multiple EndpointSlice objects, selected by labels, which must be joined to produce the full set of endpoints.

Type

**object**

## 1.11. EGRESSROUTER [NETWORK.OPERATOR.OPENSIFT.IO/V1]

### Description

EgressRouter is a feature allowing the user to define an egress router that acts as a bridge between pods and external systems. The egress router runs a service that redirects egress traffic originating from a pod or a group of pods to a remote external system or multiple destinations as per configuration.

It is consumed by the cluster-network-operator. More specifically, given an EgressRouter CR with <name>, the CNO will create and manage: - A service called <name> - An egress pod called <name> - A NAD called <name>

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

EgressRouter is a single egressrouter pod configuration object.

Type

**object**

## 1.12. INGRESS [NETWORKING.K8S.IO/V1]

### Description

Ingress is a collection of rules that allow inbound connections to reach the endpoints defined by a backend. An Ingress can be configured to give services externally-reachable urls, load balance traffic, terminate SSL, offer name based virtual hosting etc.

Type

**object**

## 1.13. INGRESSCLASS [NETWORKING.K8S.IO/V1]

### Description

IngressClass represents the class of the Ingress, referenced by the Ingress Spec. The **ingressclass.kubernetes.io/is-default-class** annotation can be used to indicate that an IngressClass should be considered default. When a single IngressClass resource has this annotation

set to true, new Ingress resources without a class specified will be assigned this default class.

Type

**object**

## 1.14. IPPOL [WHEREABOUTS.CNI.CNCF.IO/V1ALPHA1]

Description

IPPool is the Schema for the ippools API

Type

**object**

## 1.15. MULTINETWORKPOLICY [K8S.CNI.CNCF.IO/V1BETA1]

Description

MultiNetworkPolicy is a CRD schema to provide NetworkPolicy mechanism for net-attach-def which is specified by the Network Plumbing Working Group. MultiNetworkPolicy is identical to Kubernetes NetworkPolicy, See: <https://kubernetes.io/docs/concepts/services-networking/network-policies/> .

Type

**object**

## 1.16. NETWORKATTACHMENTDEFINITION [K8S.CNI.CNCF.IO/V1]

Description

NetworkAttachmentDefinition is a CRD schema specified by the Network Plumbing Working Group to express the intent for attaching pods to one or more logical or physical networks. More information available at: <https://github.com/k8snetworkplumbingwg/multi-net-spec>

Type

**object**

## 1.17. NETWORKPOLICY [NETWORKING.K8S.IO/V1]

Description

NetworkPolicy describes what network traffic is allowed for a set of Pods

Type

**object**

## 1.18. OVERLAPPINGRANGEIPRESERVATION [WHEREABOUTS.CNI.CNCF.IO/V1ALPHA1]

Description

OverlappingRangeIPReservation is the Schema for the OverlappingRangeIPReservations API

Type

**object**

## 1.19. PODNETWORKCONNECTIVITYCHECK [CONTROLPLANE.OPERATOR.OPENSIFT.IO/V1ALPHA1]

**Description**

PodNetworkConnectivityCheck

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**

## 1.20. ROUTE [ROUTE.OPENSIFT.IO/V1]

**Description**

A route allows developers to expose services through an HTTP(S) aware load balancing and proxy layer via a public DNS entry. The route may further specify TLS options and a certificate, or specify a public CNAME that the router should also accept for HTTP and HTTPS traffic. An administrator typically configures their router to be visible outside the cluster firewall, and may also add additional security, caching, or traffic controls on the service content. Routers usually talk directly to the service endpoints.

Once a route is created, the **host** field may not be changed. Generally, routers use the oldest route with a given host when resolving conflicts.

Routers are subject to additional customization and may support additional controls via the annotations field.

Because administrators may configure multiple routers, the route status field is used to return information to clients about the names and states of the route under each router. If a client chooses a duplicate name, for instance, the route status conditions are used to indicate the route cannot be chosen.

To enable HTTP/2 ALPN on a route it requires a custom (non-wildcard) certificate. This prevents connection coalescing by clients, notably web browsers. We do not support HTTP/2 ALPN on routes that use the default certificate because of the risk of connection re-use/coalescing. Routes that do not have their own custom certificate will not be HTTP/2 ALPN-enabled on either the frontend or the backend.

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

**Type**

**object**

## 1.21. SERVICE [V1]

**Description**

Service is a named abstraction of software service (for example, mysql) consisting of local port (for example 3306) that the proxy listens on, and the selector that determines which pods will answer requests sent through the proxy.

**Type**

**object**

## CHAPTER 2. ADMINNETWORKPOLICY [POLICY.NETWORKING.K8S.IO/V1ALPHA1]

### Description

AdminNetworkPolicy is a cluster level resource that is part of the AdminNetworkPolicy API.

### Type

**object**

### Required

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

## 2.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	Specification of the desired behavior of AdminNetworkPolicy.

Property	Type	Description
<b>status</b>	<b>object</b>	Status is the status to be reported by the implementation.

### 2.1.1. .spec

#### Description

Specification of the desired behavior of AdminNetworkPolicy.

#### Type

**object**

#### Required

- **priority**
- **subject**

Property	Type	Description
<b>egress</b>	<b>array</b>	<p>Egress is the list of Egress rules to be applied to the selected pods. A total of 100 rules will be allowed in each ANP instance. The relative precedence of egress rules within a single ANP object (all of which share the priority) will be determined by the order in which the rule is written. Thus, a rule that appears at the top of the egress rules would take the highest precedence. ANPs with no egress rules do not affect egress traffic.</p> <p>Support: Core</p>
<b>egress[]</b>	<b>object</b>	<p>AdminNetworkPolicyEgressRule describes an action to take on a particular set of traffic originating from pods selected by a AdminNetworkPolicy's Subject field. &lt;network-policy-api:experimental:validation&gt;</p>

Property	Type	Description
<b>ingress</b>	<b>array</b>	<p>Ingress is the list of Ingress rules to be applied to the selected pods. A total of 100 rules will be allowed in each ANP instance. The relative precedence of ingress rules within a single ANP object (all of which share the priority) will be determined by the order in which the rule is written. Thus, a rule that appears at the top of the ingress rules would take the highest precedence. ANPs with no ingress rules do not affect ingress traffic.</p> <p>Support: Core</p>
<b>ingress[]</b>	<b>object</b>	<p>AdminNetworkPolicyIngressRule describes an action to take on a particular set of traffic destined for pods selected by an AdminNetworkPolicy's Subject field.</p>
<b>priority</b>	<b>integer</b>	<p>Priority is a value from 0 to 1000. Rules with lower priority values have higher precedence, and are checked before rules with higher priority values. All AdminNetworkPolicy rules have higher precedence than NetworkPolicy or BaselineAdminNetworkPolicy rules. The behavior is undefined if two ANP objects have same priority.</p> <p>Support: Core</p>
<b>subject</b>	<b>object</b>	<p>Subject defines the pods to which this AdminNetworkPolicy applies. Note that host-networked pods are not included in subject selection.</p> <p>Support: Core</p>

### 2.1.2. .spec.egress

#### Description

Egress is the list of Egress rules to be applied to the selected pods. A total of 100 rules will be allowed

in each ANP instance. The relative precedence of egress rules within a single ANP object (all of which share the priority) will be determined by the order in which the rule is written. Thus, a rule that appears at the top of the egress rules would take the highest precedence. ANPs with no egress rules do not affect egress traffic.

Support: Core

#### Type

**array**

### 2.1.3. .spec.egress[]

#### Description

AdminNetworkPolicyEgressRule describes an action to take on a particular set of traffic originating from pods selected by a AdminNetworkPolicy's Subject field. <network-policy-api:experimental:validation>

#### Type

**object**

#### Required

- **action**
- **to**

Property	Type	Description
<b>action</b>	<b>string</b>	<p>Action specifies the effect this rule will have on matching traffic. Currently the following actions are supported: Allow: allows the selected traffic (even if it would otherwise have been denied by NetworkPolicy) Deny: denies the selected traffic Pass: instructs the selected traffic to skip any remaining ANP rules, and then pass execution to any NetworkPolicies that select the pod. If the pod is not selected by any NetworkPolicies then execution is passed to any BaselineAdminNetworkPolicies that select the pod.</p> <p>Support: Core</p>



Property	Type	Description
<b>name</b>	<b>string</b>	<p>Name is an identifier for this rule, that may be no more than 100 characters in length. This field should be used by the implementation to help improve observability, readability and error-reporting for any applied AdminNetworkPolicies.</p> <p>Support: Core</p>
<b>ports</b>	<b>array</b>	<p>Ports allows for matching traffic based on port and protocols. This field is a list of destination ports for the outgoing egress traffic. If Ports is not set then the rule does not filter traffic via port.</p> <p>Support: Core</p>
<b>ports[]</b>	<b>object</b>	<p>AdminNetworkPolicyPort describes how to select network ports on pod(s). Exactly one field must be set.</p>
<b>to</b>	<b>array</b>	<p>To is the List of destinations whose traffic this rule applies to. If any AdminNetworkPolicyEgressPeer matches the destination of outgoing traffic then the specified action is applied. This field must be defined and contain at least one item.</p> <p>Support: Core</p>
<b>to[]</b>	<b>object</b>	<p>AdminNetworkPolicyEgressPeer defines a peer to allow traffic to. Exactly one of the selector pointers must be set for a given peer. If a consumer observes none of its fields are set, they must assume an unknown option has been specified and fail closed.</p>

#### 2.1.4. .spec.egress[].ports

### Description

Ports allows for matching traffic based on port and protocols. This field is a list of destination ports for the outgoing egress traffic. If Ports is not set then the rule does not filter traffic via port.  
Support: Core

### Type

**array**

## 2.1.5. .spec.egress[].ports[]

### Description

AdminNetworkPolicyPort describes how to select network ports on pod(s). Exactly one field must be set.

### Type

**object**

Property	Type	Description
<b>namedPort</b>	<b>string</b>	NamedPort selects a port on a pod(s) based on name.  Support: Extended  <network-policy-api:experimental>
<b>portNumber</b>	<b>object</b>	Port selects a port on a pod(s) based on number.  Support: Core
<b>portRange</b>	<b>object</b>	PortRange selects a port range on a pod(s) based on provided start and end values.  Support: Core

## 2.1.6. .spec.egress[].ports[].portNumber

### Description

Port selects a port on a pod(s) based on number.  
Support: Core

### Type

**object**

### Required

- **port**
- **protocol**

Property	Type	Description
<b>port</b>	<b>integer</b>	Number defines a network port value.  Support: Core
<b>protocol</b>	<b>string</b>	Protocol is the network protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.  Support: Core

### 2.1.7. .spec.egress[].ports[].portRange

#### Description

PortRange selects a port range on a pod(s) based on provided start and end values.

Support: Core

#### Type

**object**

#### Required

- **end**
- **start**

Property	Type	Description
<b>end</b>	<b>integer</b>	End defines a network port that is the end of a port range, the End value must be greater than Start.  Support: Core
<b>protocol</b>	<b>string</b>	Protocol is the network protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.  Support: Core
<b>start</b>	<b>integer</b>	Start defines a network port that is the start of a port range, the Start value must be less than End.  Support: Core

### 2.1.8. .spec.egress[].to

#### Description

To is the List of destinations whose traffic this rule applies to. If any AdminNetworkPolicyEgressPeer matches the destination of outgoing traffic then the specified action is applied. This field must be defined and contain at least one item.

Support: Core

#### Type

**array**

### 2.1.9. .spec.egress[].to[]

#### Description

AdminNetworkPolicyEgressPeer defines a peer to allow traffic to. Exactly one of the selector pointers must be set for a given peer. If a consumer observes none of its fields are set, they must assume an unknown option has been specified and fail closed.

#### Type

**object**

Property	Type	Description
<b>namespaces</b>	<b>object</b>	<p>Namespaces defines a way to select all pods within a set of Namespaces. Note that host-networked pods are not included in this type of peer.</p> <p>Support: Core</p>

Property	Type	Description
<b>networks</b>	<b>array (string)</b>	<p>Networks defines a way to select peers via CIDR blocks. This is intended for representing entities that live outside the cluster, which can't be selected by pods, namespaces and nodes peers, but note that cluster-internal traffic will be checked against the rule as well. So if you Allow or Deny traffic to <b>"0.0.0.0/0"</b>, that will allow or deny all IPv4 pod-to-pod traffic as well. If you don't want that, add a rule that Passes all pod traffic before the Networks rule.</p> <p>Each item in Networks should be provided in the CIDR format and should be IPv4 or IPv6, for example "10.0.0.0/8" or "fd00::/8".</p> <p>Networks can have upto 25 CIDRs specified.</p> <p>Support: Extended</p> <p>&lt;network-policy-api:experimental&gt;</p>
<b>nodes</b>	<b>object</b>	<p>Nodes defines a way to select a set of nodes in the cluster. This field follows standard label selector semantics; if present but empty, it selects all Nodes.</p> <p>Support: Extended</p> <p>&lt;network-policy-api:experimental&gt;</p>
<b>pods</b>	<b>object</b>	<p>Pods defines a way to select a set of pods in a set of namespaces. Note that host-networked pods are not included in this type of peer.</p> <p>Support: Core</p>

### 2.1.10. .spec.egress[].to[].namespaces

#### Description

Namespaces defines a way to select all pods within a set of Namespaces. Note that host-networked pods are not included in this type of peer.

Support: Core

## Type

**object**

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

### 2.1.11. .spec.egress[].to[].namespaces.matchExpressions

#### Description

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

#### Type

**array**

### 2.1.12. .spec.egress[].to[].namespaces.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

### 2.1.13. .spec.egress[].to[].nodes

#### Description

Nodes defines a way to select a set of nodes in the cluster. This field follows standard label selector semantics; if present but empty, it selects all Nodes.

Support: Extended

<network-policy-api:experimental>

#### Type

**object**

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

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

#### 2.1.14. .spec.egress[.to[].nodes.matchExpressions

##### Description

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

##### Type

**array**

#### 2.1.15. .spec.egress[.to[].nodes.matchExpressions[.]

##### Description

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

##### Type

**object**

##### Required

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

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



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

### 2.1.16. .spec.egress[].to[].pods

#### Description

Pods defines a way to select a set of pods in a set of namespaces. Note that host-networked pods are not included in this type of peer.

Support: Core

#### Type

**object**

#### Required

- **namespaceSelector**
- **podSelector**

Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.
<b>podSelector</b>	<b>object</b>	PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

### 2.1.17. .spec.egress[].to[].pods.namespaceSelector

#### Description

NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.

#### Type

**object**

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

### 2.1.18. .spec.egress[].to[].pods.namespaceSelector.matchExpressions

#### Description

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

#### Type

**array**

### 2.1.19. .spec.egress[].to[].pods.namespaceSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

Property	Type	Description
<b>key</b>	<b>string</b>	key is the label key that the selector applies to.

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

### 2.1.20. .spec.egress[].to[].pods.podSelector

#### Description

PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

#### Type

**object**

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

### 2.1.21. .spec.egress[].to[].pods.podSelector.matchExpressions

#### Description

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

#### Type

**array****2.1.22. .spec.egress[].to[].pods.podSelector.matchExpressions[]****Description**

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

**Type****object****Required**

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

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

**2.1.23. .spec.ingress****Description**

Ingress is the list of Ingress rules to be applied to the selected pods. A total of 100 rules will be allowed in each ANP instance. The relative precedence of ingress rules within a single ANP object (all of which share the priority) will be determined by the order in which the rule is written. Thus, a rule that appears at the top of the ingress rules would take the highest precedence. ANPs with no ingress rules do not affect ingress traffic.

Support: Core

**Type****array****2.1.24. .spec.ingress[]**

**Description**

AdminNetworkPolicyIngressRule describes an action to take on a particular set of traffic destined for pods selected by an AdminNetworkPolicy's Subject field.

**Type**

**object**

**Required**

- **action**
- **from**

Property	Type	Description
<b>action</b>	<b>string</b>	<p>Action specifies the effect this rule will have on matching traffic. Currently the following actions are supported: Allow: allows the selected traffic (even if it would otherwise have been denied by NetworkPolicy) Deny: denies the selected traffic Pass: instructs the selected traffic to skip any remaining ANP rules, and then pass execution to any NetworkPolicies that select the pod. If the pod is not selected by any NetworkPolicies then execution is passed to any BaselineAdminNetworkPolicies that select the pod.</p> <p>Support: Core</p>
<b>from</b>	<b>array</b>	<p>From is the list of sources whose traffic this rule applies to. If any AdminNetworkPolicyIngressPeer matches the source of incoming traffic then the specified action is applied. This field must be defined and contain at least one item.</p> <p>Support: Core</p>

Property	Type	Description
<b>from[]</b>	<b>object</b>	AdminNetworkPolicyIngressPeer defines an in-cluster peer to allow traffic from. Exactly one of the selector pointers must be set for a given peer. If a consumer observes none of its fields are set, they must assume an unknown option has been specified and fail closed.
<b>name</b>	<b>string</b>	Name is an identifier for this rule, that may be no more than 100 characters in length. This field should be used by the implementation to help improve observability, readability and error-reporting for any applied AdminNetworkPolicies.  Support: Core
<b>ports</b>	<b>array</b>	Ports allows for matching traffic based on port and protocols. This field is a list of ports which should be matched on the pods selected for this policy i.e the subject of the policy. So it matches on the destination port for the ingress traffic. If Ports is not set then the rule does not filter traffic via port.  Support: Core
<b>ports[]</b>	<b>object</b>	AdminNetworkPolicyPort describes how to select network ports on pod(s). Exactly one field must be set.

### 2.1.25. .spec.ingress[].from

#### Description

From is the list of sources whose traffic this rule applies to. If any AdminNetworkPolicyIngressPeer matches the source of incoming traffic then the specified action is applied. This field must be defined and contain at least one item.

Support: Core

#### Type

**array**

### 2.1.26. .spec.ingress[].from[]

#### Description

AdminNetworkPolicyIngressPeer defines an in-cluster peer to allow traffic from. Exactly one of the selector pointers must be set for a given peer. If a consumer observes none of its fields are set, they must assume an unknown option has been specified and fail closed.

#### Type

**object**

Property	Type	Description
<b>namespaces</b>	<b>object</b>	Namespaces defines a way to select all pods within a set of Namespaces. Note that host-networked pods are not included in this type of peer.  Support: Core
<b>pods</b>	<b>object</b>	Pods defines a way to select a set of pods in a set of namespaces. Note that host-networked pods are not included in this type of peer.  Support: Core

### 2.1.27. .spec.ingress[].from[].namespaces

#### Description

Namespaces defines a way to select all pods within a set of Namespaces. Note that host-networked pods are not included in this type of peer.

Support: Core

#### Type

**object**

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

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

### 2.1.28. .spec.ingress[].from[].namespaces.matchExpressions

#### Description

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

#### Type

**array**

### 2.1.29. .spec.ingress[].from[].namespaces.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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



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

### 2.1.30. .spec.ingress[].from[].pods

#### Description

Pods defines a way to select a set of pods in a set of namespaces. Note that host-networked pods are not included in this type of peer.

Support: Core

#### Type

**object**

#### Required

- **namespaceSelector**
- **podSelector**

Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.
<b>podSelector</b>	<b>object</b>	PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

### 2.1.31. .spec.ingress[].from[].pods.namespaceSelector

#### Description

NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.

#### Type

**object**

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

### 2.1.32. .spec.ingress[].from[].pods.namespaceSelector.matchExpressions

#### Description

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

#### Type

**array**

### 2.1.33. .spec.ingress[].from[].pods.namespaceSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

Property	Type	Description
<b>key</b>	<b>string</b>	key is the label key that the selector applies to.

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

### 2.1.34. .spec.ingress[].from[].pods.podSelector

#### Description

PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

#### Type

**object**

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

### 2.1.35. .spec.ingress[].from[].pods.podSelector.matchExpressions

#### Description

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

Type

**array**

### 2.1.36. .spec.ingress[].from[].pods.podSelector.matchExpressions[]

Description

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

Type

**object**

Required

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

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

### 2.1.37. .spec.ingress[].ports

Description

Ports allows for matching traffic based on port and protocols. This field is a list of ports which should be matched on the pods selected for this policy i.e the subject of the policy. So it matches on the destination port for the ingress traffic. If Ports is not set then the rule does not filter traffic via port. Support: Core

Type

**array**

### 2.1.38. .spec.ingress[].ports[]

**Description**

AdminNetworkPolicyPort describes how to select network ports on pod(s). Exactly one field must be set.

**Type**

**object**

Property	Type	Description
<b>namedPort</b>	<b>string</b>	NamedPort selects a port on a pod(s) based on name.  Support: Extended  <network-policy-api:experimental>
<b>portNumber</b>	<b>object</b>	Port selects a port on a pod(s) based on number.  Support: Core
<b>portRange</b>	<b>object</b>	PortRange selects a port range on a pod(s) based on provided start and end values.  Support: Core

**2.1.39. .spec.ingress[].ports[].portNumber****Description**

Port selects a port on a pod(s) based on number.  
Support: Core

**Type**

**object**

**Required**

- **port**
- **protocol**

Property	Type	Description
<b>port</b>	<b>integer</b>	Number defines a network port value.  Support: Core

Property	Type	Description
<b>protocol</b>	<b>string</b>	<p>Protocol is the network protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.</p> <p>Support: Core</p>

### 2.1.40. .spec.ingress[].ports[].portRange

#### Description

PortRange selects a port range on a pod(s) based on provided start and end values.

Support: Core

#### Type

**object**

#### Required

- **end**
- **start**

Property	Type	Description
<b>end</b>	<b>integer</b>	<p>End defines a network port that is the end of a port range, the End value must be greater than Start.</p> <p>Support: Core</p>
<b>protocol</b>	<b>string</b>	<p>Protocol is the network protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.</p> <p>Support: Core</p>
<b>start</b>	<b>integer</b>	<p>Start defines a network port that is the start of a port range, the Start value must be less than End.</p> <p>Support: Core</p>

### 2.1.41. .spec.subject

**Description**

Subject defines the pods to which this AdminNetworkPolicy applies. Note that host-networked pods are not included in subject selection.

Support: Core

**Type**

**object**

Property	Type	Description
<b>namespaces</b>	<b>object</b>	Namespaces is used to select pods via namespace selectors.
<b>pods</b>	<b>object</b>	Pods is used to select pods via namespace AND pod selectors.

**2.1.42. .spec.subject.namespaces****Description**

Namespaces is used to select pods via namespace selectors.

**Type**

**object**

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

**2.1.43. .spec.subject.namespaces.matchExpressions****Description**

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

**Type****array****2.1.44. .spec.subject.namespaces.matchExpressions[]****Description**

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

**Type****object****Required**

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

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

**2.1.45. .spec.subject.pods****Description**

Pods is used to select pods via namespace AND pod selectors.

**Type****object****Required**

- **namespaceSelector**
- **podSelector**



Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.
<b>podSelector</b>	<b>object</b>	PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

### 2.1.46. .spec.subject.pods.namespaceSelector

#### Description

NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.

#### Type

**object**

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

### 2.1.47. .spec.subject.pods.namespaceSelector.matchExpressions

#### Description

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

#### Type

**array**

### 2.1.48. .spec.subject.pods.namespaceSelector.matchExpressions[]

**Description**

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

**Type**

**object**

**Required**

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

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

**2.1.49. .spec.subject.pods.podSelector****Description**

PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

**Type**

**object**

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

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

### 2.1.50. .spec.subject.pods.podSelector.matchExpressions

#### Description

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

#### Type

**array**

### 2.1.51. .spec.subject.pods.podSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

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

## 2.1.52. .status

### Description

Status is the status to be reported by the implementation.

### Type

**object**

### Required

- **conditions**

Property	Type	Description
<b>conditions</b>	<b>array</b>	

Property	Type	Description
<b>conditions[]</b>	<b>object</b>	<p>Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example,</p> <pre> type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition <b>json:"conditions,omitempty"</b> <b>patchStrategy:"merge"</b> <b>patchMergeKey:"type"</b> <b>protobuf:"bytes,1,rep,name=conditions"</b>  // other fields }</pre>

### 2.1.53. .status.conditions

#### Description

#### Type

**array**

### 2.1.54. .status.conditions[]

#### Description

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

```

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

```

```

    // other fields
  }

```

**Type****object****Required**

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

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

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

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

## 2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies**
  - **DELETE:** delete collection of AdminNetworkPolicy
  - **GET:** list objects of kind AdminNetworkPolicy
  - **POST:** create an AdminNetworkPolicy
- **/apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies/{name}**
  - **DELETE:** delete an AdminNetworkPolicy
  - **GET:** read the specified AdminNetworkPolicy
  - **PATCH:** partially update the specified AdminNetworkPolicy
  - **PUT:** replace the specified AdminNetworkPolicy
- **/apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies/{name}/status**
  - **GET:** read status of the specified AdminNetworkPolicy

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

### 2.2.1. /apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies

#### HTTP method

##### DELETE

#### Description

delete collection of AdminNetworkPolicy

Table 2.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

##### GET

#### Description

list objects of kind AdminNetworkPolicy

Table 2.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">AdminNetworkPolicyList</a> schema
401 - Unauthorized	Empty

#### HTTP method

##### POST

#### Description

create an AdminNetworkPolicy

Table 2.3. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 2.4. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">AdminNetworkPolicy</a> schema	

Table 2.5. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">AdminNetworkPolicy</a> schema
201 - Created	<a href="#">AdminNetworkPolicy</a> schema
202 - Accepted	<a href="#">AdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

### 2.2.2. /apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies/{name}

Table 2.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the AdminNetworkPolicy

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

delete an AdminNetworkPolicy

**Table 2.7. Query parameters**

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

**Table 2.8. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

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

read the specified AdminNetworkPolicy

**Table 2.9. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">AdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

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

partially update the specified AdminNetworkPolicy

**Table 2.10. Query parameters**

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

Table 2.11. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">AdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

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

replace the specified AdminNetworkPolicy

Table 2.12. Query parameters

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

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

Table 2.13. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">AdminNetworkPolicy</a> schema	

Table 2.14. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">AdminNetworkPolicy</a> schema
201 - Created	<a href="#">AdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

### 2.2.3. /apis/policy.networking.k8s.io/v1alpha1/adminnetworkpolicies/{name}/status

Table 2.15. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the AdminNetworkPolicy

HTTP method

**GET**

Description

read status of the specified AdminNetworkPolicy

Table 2.16. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">AdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

HTTP method

**PATCH**

Description

partially update status of the specified AdminNetworkPolicy

Table 2.17. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 2.18. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">AdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

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

replace status of the specified AdminNetworkPolicy

Table 2.19. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 2.20. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">AdminNetworkPolicy</a> schema	

Table 2.21. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">AdminNetworkPolicy</a> schema
201 - Created	<a href="#">AdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

## CHAPTER 3. ADMINPOLICYBASEDEXTERNALROUTE [K8S.OVN.ORG/V1]

### Description

AdminPolicyBasedExternalRoute is a CRD allowing the cluster administrators to configure policies for external gateway IPs to be applied to all the pods contained in selected namespaces. Egress traffic from the pods that belong to the selected namespaces to outside the cluster is routed through these external gateway IPs.

### Type

**object**

### Required

- **spec**

## 3.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>



Property	Type	Description
<b>spec</b>	<b>object</b>	AdminPolicyBasedExternalRouteSpec defines the desired state of AdminPolicyBasedExternalRoute
<b>status</b>	<b>object</b>	AdminPolicyBasedRouteStatus contains the observed status of the AdminPolicyBased route types.

### 3.1.1. .spec

#### Description

AdminPolicyBasedExternalRouteSpec defines the desired state of AdminPolicyBasedExternalRoute

#### Type

**object**

#### Required

- **from**
- **nextHops**

Property	Type	Description
<b>from</b>	<b>object</b>	From defines the selectors that will determine the target namespaces to this CR.
<b>nextHops</b>	<b>object</b>	NextHops defines two types of hops: Static and Dynamic. Each hop defines at least one external gateway IP.

### 3.1.2. .spec.from

#### Description

From defines the selectors that will determine the target namespaces to this CR.

#### Type

**object**

#### Required

- **namespaceSelector**

Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector defines a selector to be used to determine which namespaces will be targeted by this CR

### 3.1.3. .spec.from.namespaceSelector

#### Description

NamespaceSelector defines a selector to be used to determine which namespaces will be targeted by this CR

#### Type

**object**

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

### 3.1.4. .spec.from.namespaceSelector.matchExpressions

#### Description

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

#### Type

**array**

### 3.1.5. .spec.from.namespaceSelector.matchExpressions[]

#### Description

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

**Type****object****Required**

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

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

### 3.1.6. .spec.nextHops

**Description**

NextHops defines two types of hops: Static and Dynamic. Each hop defines at least one external gateway IP.

**Type****object**

Property	Type	Description
<b>dynamic</b>	<b>array</b>	DynamicHops defines a slices of DynamicHop. This field is optional.

Property	Type	Description
<b>dynamic[]</b>	<b>object</b>	DynamicHop defines the configuration for a dynamic external gateway interface. These interfaces are wrapped around a pod object that resides inside the cluster. The field NetworkAttachmentName captures the name of the multus network name to use when retrieving the gateway IP to use. The PodSelector and the NamespaceSelector are mandatory fields.
<b>static</b>	<b>array</b>	StaticHops defines a slice of StaticHop. This field is optional.
<b>static[]</b>	<b>object</b>	StaticHop defines the configuration of a static IP that acts as an external Gateway Interface. IP field is mandatory.

### 3.1.7. .spec.nextHops.dynamic

#### Description

DynamicHops defines a slices of DynamicHop. This field is optional.

#### Type

**array**

### 3.1.8. .spec.nextHops.dynamic[]

#### Description

DynamicHop defines the configuration for a dynamic external gateway interface. These interfaces are wrapped around a pod object that resides inside the cluster. The field NetworkAttachmentName captures the name of the multus network name to use when retrieving the gateway IP to use. The PodSelector and the NamespaceSelector are mandatory fields.

#### Type

**object**

#### Required

- **namespaceSelector**
- **podSelector**

Property	Type	Description
<b>bfdEnabled</b>	<b>boolean</b>	BFDEnabled determines if the interface implements the Bidirectional Forward Detection protocol. Defaults to false.
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector defines a selector to filter the namespaces where the pod gateways are located.
<b>networkAttachmentName</b>	<b>string</b>	NetworkAttachmentName determines the multus network name to use when retrieving the pod IPs that will be used as the gateway IP. When this field is empty, the logic assumes that the pod is configured with HostNetwork and is using the node's IP as gateway.
<b>podSelector</b>	<b>object</b>	PodSelector defines the selector to filter the pods that are external gateways.

### 3.1.9. .spec.nextHops.dynamic[].namespaceSelector

#### Description

NamespaceSelector defines a selector to filter the namespaces where the pod gateways are located.

#### Type

**object**

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

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

### 3.1.10. .spec.nextHops.dynamic[].namespaceSelector.matchExpressions

#### Description

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

#### Type

**array**

### 3.1.11. .spec.nextHops.dynamic[].namespaceSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

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

### 3.1.12. .spec.nextHops.dynamic[].podSelector

#### Description

PodSelector defines the selector to filter the pods that are external gateways.

#### Type

**object**

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

### 3.1.13. .spec.nextHops.dynamic[].podSelector.matchExpressions

#### Description

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

#### Type

**array**

### 3.1.14. .spec.nextHops.dynamic[].podSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

### 3.1.15. .spec.nextHops.static

#### Description

StaticHops defines a slice of StaticHop. This field is optional.

#### Type

**array**

### 3.1.16. .spec.nextHops.static[]

#### Description

StaticHop defines the configuration of a static IP that acts as an external Gateway Interface. IP field is mandatory.

#### Type

**object**

#### Required

- **ip**



Property	Type	Description
<b>bfdEnabled</b>	<b>boolean</b>	BFDEnabled determines if the interface implements the Bidirectional Forward Detection protocol. Defaults to false.
<b>ip</b>	<b>string</b>	IP defines the static IP to be used for egress traffic. The IP can be either IPv4 or IPv6.

### 3.1.17. .status

#### Description

AdminPolicyBasedRouteStatus contains the observed status of the AdminPolicyBased route types.

#### Type

**object**

Property	Type	Description
<b>lastTransitionTime</b>	<b>string</b>	Captures the time when the last change was applied.
<b>messages</b>	<b>array (string)</b>	An array of Human-readable messages indicating details about the status of the object.
<b>status</b>	<b>string</b>	A concise indication of whether the AdminPolicyBasedRoute resource is applied with success

## 3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes**
  - **DELETE**: delete collection of AdminPolicyBasedExternalRoute
  - **GET**: list objects of kind AdminPolicyBasedExternalRoute
  - **POST**: create an AdminPolicyBasedExternalRoute
- **/apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes/{name}**
  - **DELETE**: delete an AdminPolicyBasedExternalRoute
  - **GET**: read the specified AdminPolicyBasedExternalRoute
  - **PATCH**: partially update the specified AdminPolicyBasedExternalRoute

- **PUT**: replace the specified AdminPolicyBasedExternalRoute
- **/apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes/{name}/status**
  - **GET**: read status of the specified AdminPolicyBasedExternalRoute
  - **PATCH**: partially update status of the specified AdminPolicyBasedExternalRoute
  - **PUT**: replace status of the specified AdminPolicyBasedExternalRoute

### 3.2.1. /apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes

#### HTTP method

#### DELETE

#### Description

delete collection of AdminPolicyBasedExternalRoute

Table 3.1. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

list objects of kind AdminPolicyBasedExternalRoute

Table 3.2. HTTP responses

HTTP code	Reponse body
200 - OK	<b>AdminPolicyBasedExternalRouteList</b> schema
401 - Unauthorized	Empty

#### HTTP method

#### POST

#### Description

create an AdminPolicyBasedExternalRoute

Table 3.3. Query parameters

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

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

Table 3.4. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">AdminPolicyBasedExternalRoute</a> schema	

Table 3.5. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">AdminPolicyBasedExternalRoute</a> schema
201 - Created	<a href="#">AdminPolicyBasedExternalRoute</a> schema
202 - Accepted	<a href="#">AdminPolicyBasedExternalRoute</a> schema
401 - Unauthorized	Empty

### 3.2.2. /apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes/{name}

Table 3.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the AdminPolicyBasedExternalRoute

#### HTTP method

#### DELETE

#### Description

delete an AdminPolicyBasedExternalRoute

Table 3.7. Query parameters

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

Table 3.8. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

read the specified AdminPolicyBasedExternalRoute

Table 3.9. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">AdminPolicyBasedExternalRoute</a> schema
401 - Unauthorized	Empty

#### HTTP method

**PATCH****Description**

partially update the specified AdminPolicyBasedExternalRoute

**Table 3.10. Query parameters**

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

**Table 3.11. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">AdminPolicyBasedExternalRoute</a> schema
401 - Unauthorized	Empty

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

replace the specified AdminPolicyBasedExternalRoute

**Table 3.12. Query parameters**

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

Table 3.13. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">AdminPolicyBasedExternalRoute</a> schema	

Table 3.14. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">AdminPolicyBasedExternalRoute</a> schema
201 - Created	<a href="#">AdminPolicyBasedExternalRoute</a> schema
401 - Unauthorized	Empty

### 3.2.3. /apis/k8s.ovn.org/v1/adminpolicybasedexternalroutes/{name}/status

Table 3.15. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the AdminPolicyBasedExternalRoute

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

read status of the specified AdminPolicyBasedExternalRoute

**Table 3.16. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">AdminPolicyBasedExternalRoute</a> schema
401 - Unauthorized	Empty

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

partially update status of the specified AdminPolicyBasedExternalRoute

**Table 3.17. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 3.18. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">AdminPolicyBasedExternalRoute</a> schema
401 - Unauthorized	Empty

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

replace status of the specified AdminPolicyBasedExternalRoute

Table 3.19. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>



Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 3.20. Body parameters

Parameter	Type	Description
<b>body</b>	<b><a href="#">AdminPolicyBasedExternalRoute</a></b> schema	

Table 3.21. HTTP responses

HTTP code	Reponse body
200 - OK	<b><a href="#">AdminPolicyBasedExternalRoute</a></b> schema
201 - Created	<b><a href="#">AdminPolicyBasedExternalRoute</a></b> schema
401 - Unauthorized	Empty

## CHAPTER 4. BASELINEADMINNETWORKPOLICY [POLICY.NETWORKING.K8S.IO/V1ALPHA1]

### Description

BaselineAdminNetworkPolicy is a cluster level resource that is part of the AdminNetworkPolicy API.

### Type

**object**

### Required

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

## 4.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	Specification of the desired behavior of BaselineAdminNetworkPolicy.

Property	Type	Description
<b>status</b>	<b>object</b>	Status is the status to be reported by the implementation.

### 4.1.1. .spec

#### Description

Specification of the desired behavior of BaselineAdminNetworkPolicy.

#### Type

**object**

#### Required

- **subject**

Property	Type	Description
<b>egress</b>	<b>array</b>	<p>Egress is the list of Egress rules to be applied to the selected pods if they are not matched by any AdminNetworkPolicy or NetworkPolicy rules. A total of 100 Egress rules will be allowed in each BANP instance. The relative precedence of egress rules within a single BANP object will be determined by the order in which the rule is written. Thus, a rule that appears at the top of the egress rules would take the highest precedence. BANPs with no egress rules do not affect egress traffic.</p> <p>Support: Core</p>
<b>egress[]</b>	<b>object</b>	<p>BaselineAdminNetworkPolicyEgressRule describes an action to take on a particular set of traffic originating from pods selected by a BaselineAdminNetworkPolicy's Subject field. &lt;network-policy-api:experimental:validation&gt;</p>

Property	Type	Description
<b>ingress</b>	<b>array</b>	<p>Ingress is the list of Ingress rules to be applied to the selected pods if they are not matched by any AdminNetworkPolicy or NetworkPolicy rules. A total of 100 Ingress rules will be allowed in each BANP instance. The relative precedence of ingress rules within a single BANP object will be determined by the order in which the rule is written. Thus, a rule that appears at the top of the ingress rules would take the highest precedence. BANPs with no ingress rules do not affect ingress traffic.</p> <p>Support: Core</p>
<b>ingress[]</b>	<b>object</b>	<p>BaselineAdminNetworkPolicyIngressRule describes an action to take on a particular set of traffic destined for pods selected by a BaselineAdminNetworkPolicy's Subject field.</p>
<b>subject</b>	<b>object</b>	<p>Subject defines the pods to which this BaselineAdminNetworkPolicy applies. Note that host-networked pods are not included in subject selection.</p> <p>Support: Core</p>

### 4.1.2. .spec.egress

#### Description

Egress is the list of Egress rules to be applied to the selected pods if they are not matched by any AdminNetworkPolicy or NetworkPolicy rules. A total of 100 Egress rules will be allowed in each BANP instance. The relative precedence of egress rules within a single BANP object will be determined by the order in which the rule is written. Thus, a rule that appears at the top of the egress rules would take the highest precedence. BANPs with no egress rules do not affect egress traffic.

Support: Core

#### Type

**array**

### 4.1.3. .spec.egress[]

#### Description

BaselineAdminNetworkPolicyEgressRule describes an action to take on a particular set of traffic originating from pods selected by a BaselineAdminNetworkPolicy's Subject field. <network-policy-api:experimental:validation>

## Type

**object**

## Required

- **action**
- **to**

Property	Type	Description
<b>action</b>	<b>string</b>	Action specifies the effect this rule will have on matching traffic. Currently the following actions are supported: Allow: allows the selected traffic Deny: denies the selected traffic  Support: Core
<b>name</b>	<b>string</b>	Name is an identifier for this rule, that may be no more than 100 characters in length. This field should be used by the implementation to help improve observability, readability and error-reporting for any applied BaselineAdminNetworkPolicies.  Support: Core
<b>ports</b>	<b>array</b>	Ports allows for matching traffic based on port and protocols. This field is a list of destination ports for the outgoing egress traffic. If Ports is not set then the rule does not filter traffic via port.
<b>ports[]</b>	<b>object</b>	AdminNetworkPolicyPort describes how to select network ports on pod(s). Exactly one field must be set.

Property	Type	Description
<b>to</b>	<b>array</b>	<p>To is the list of destinations whose traffic this rule applies to. If any AdminNetworkPolicyEgressPeer matches the destination of outgoing traffic then the specified action is applied. This field must be defined and contain at least one item.</p> <p>Support: Core</p>
<b>to[]</b>	<b>object</b>	<p>AdminNetworkPolicyEgressPeer defines a peer to allow traffic to. Exactly one of the selector pointers must be set for a given peer. If a consumer observes none of its fields are set, they must assume an unknown option has been specified and fail closed.</p>

#### 4.1.4. .spec.egress[].ports

##### Description

Ports allows for matching traffic based on port and protocols. This field is a list of destination ports for the outgoing egress traffic. If Ports is not set then the rule does not filter traffic via port.

##### Type

**array**

#### 4.1.5. .spec.egress[].ports[]

##### Description

AdminNetworkPolicyPort describes how to select network ports on pod(s). Exactly one field must be set.

##### Type

**object**

Property	Type	Description
<b>namedPort</b>	<b>string</b>	<p>NamedPort selects a port on a pod(s) based on name.</p> <p>Support: Extended</p> <p>&lt;network-policy-api:experimental&gt;</p>

Property	Type	Description
<b>portNumber</b>	<b>object</b>	Port selects a port on a pod(s) based on number.  Support: Core
<b>portRange</b>	<b>object</b>	PortRange selects a port range on a pod(s) based on provided start and end values.  Support: Core

#### 4.1.6. .spec.egress[].ports[].portNumber

##### Description

Port selects a port on a pod(s) based on number.  
Support: Core

##### Type

**object**

##### Required

- **port**
- **protocol**

Property	Type	Description
<b>port</b>	<b>integer</b>	Number defines a network port value.  Support: Core
<b>protocol</b>	<b>string</b>	Protocol is the network protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.  Support: Core

#### 4.1.7. .spec.egress[].ports[].portRange

##### Description

PortRange selects a port range on a pod(s) based on provided start and end values.  
Support: Core

##### Type

**object****Required**

- **end**
- **start**

Property	Type	Description
<b>end</b>	<b>integer</b>	End defines a network port that is the end of a port range, the End value must be greater than Start.  Support: Core
<b>protocol</b>	<b>string</b>	Protocol is the network protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.  Support: Core
<b>start</b>	<b>integer</b>	Start defines a network port that is the start of a port range, the Start value must be less than End.  Support: Core

**4.1.8. .spec.egress[].to****Description**

To is the list of destinations whose traffic this rule applies to. If any AdminNetworkPolicyEgressPeer matches the destination of outgoing traffic then the specified action is applied. This field must be defined and contain at least one item.

Support: Core

**Type**

**array**

**4.1.9. .spec.egress[].to[]****Description**

AdminNetworkPolicyEgressPeer defines a peer to allow traffic to. Exactly one of the selector pointers must be set for a given peer. If a consumer observes none of its fields are set, they must assume an unknown option has been specified and fail closed.

**Type**

**object**



Property	Type	Description
<b>namespaces</b>	<b>object</b>	<p>Namespaces defines a way to select all pods within a set of Namespaces. Note that host-networked pods are not included in this type of peer.</p> <p>Support: Core</p>
<b>networks</b>	<b>array (string)</b>	<p>Networks defines a way to select peers via CIDR blocks. This is intended for representing entities that live outside the cluster, which can't be selected by pods, namespaces and nodes peers, but note that cluster-internal traffic will be checked against the rule as well. So if you Allow or Deny traffic to <b>"0.0.0.0/0"</b>, that will allow or deny all IPv4 pod-to-pod traffic as well. If you don't want that, add a rule that Passes all pod traffic before the Networks rule.</p> <p>Each item in Networks should be provided in the CIDR format and should be IPv4 or IPv6, for example "10.0.0.0/8" or "fd00::/8".</p> <p>Networks can have upto 25 CIDRs specified.</p> <p>Support: Extended</p> <p>&lt;network-policy-api:experimental&gt;</p>
<b>nodes</b>	<b>object</b>	<p>Nodes defines a way to select a set of nodes in the cluster. This field follows standard label selector semantics; if present but empty, it selects all Nodes.</p> <p>Support: Extended</p> <p>&lt;network-policy-api:experimental&gt;</p>

Property	Type	Description
<b>pods</b>	<b>object</b>	Pods defines a way to select a set of pods in a set of namespaces. Note that host-networked pods are not included in this type of peer.  Support: Core

#### 4.1.10. .spec.egress[].to[].namespaces

##### Description

Namespaces defines a way to select all pods within a set of Namespaces. Note that host-networked pods are not included in this type of peer.

Support: Core

##### Type

**object**

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

#### 4.1.11. .spec.egress[].to[].namespaces.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.12. .spec.egress[].to[].namespaces.matchExpressions[]

**Description**

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

**Type**

**object**

**Required**

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

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

**4.1.13. .spec.egress[].to[].nodes****Description**

Nodes defines a way to select a set of nodes in the cluster. This field follows standard label selector semantics; if present but empty, it selects all Nodes.

Support: Extended

<network-policy-api:experimental>

**Type**

**object**

Property	Type	Description
<b>matchExpressions</b>	<b>array</b>	matchExpressions is a list of label selector requirements. The requirements are ANDed.

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

#### 4.1.14. .spec.egress[].to[].nodes.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.15. .spec.egress[].to[].nodes.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

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

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

#### 4.1.16. .spec.egress[].to[].pods

##### Description

Pods defines a way to select a set of pods in a set of namespaces. Note that host-networked pods are not included in this type of peer.

Support: Core

##### Type

**object**

##### Required

- **namespaceSelector**
- **podSelector**

Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.
<b>podSelector</b>	<b>object</b>	PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

#### 4.1.17. .spec.egress[].to[].pods.namespaceSelector

##### Description

NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.

##### Type

**object**

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

#### 4.1.18. .spec.egress[].to[].pods.namespaceSelector.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.19. .spec.egress[].to[].pods.namespaceSelector.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

Property	Type	Description
<b>key</b>	<b>string</b>	key is the label key that the selector applies to.

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

#### 4.1.20. .spec.egress[].to[].pods.podSelector

##### Description

PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

##### Type

**object**

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

#### 4.1.21. .spec.egress[].to[].pods.podSelector.matchExpressions

##### Description

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

Type

**array**

#### 4.1.22. .spec.egress[].to[].pods.podSelector.matchExpressions[]

Description

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

Type

**object**

Required

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

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

#### 4.1.23. .spec.ingress

Description

Ingress is the list of Ingress rules to be applied to the selected pods if they are not matched by any AdminNetworkPolicy or NetworkPolicy rules. A total of 100 Ingress rules will be allowed in each BANP instance. The relative precedence of ingress rules within a single BANP object will be determined by the order in which the rule is written. Thus, a rule that appears at the top of the ingress rules would take the highest precedence. BANPs with no ingress rules do not affect ingress traffic.

Support: Core

Type

**array**



### 4.1.24. .spec.ingress[]

#### Description

BaselineAdminNetworkPolicyIngressRule describes an action to take on a particular set of traffic destined for pods selected by a BaselineAdminNetworkPolicy's Subject field.

#### Type

**object**

#### Required

- **action**
- **from**

Property	Type	Description
<b>action</b>	<b>string</b>	Action specifies the effect this rule will have on matching traffic. Currently the following actions are supported: Allow: allows the selected traffic Deny: denies the selected traffic  Support: Core
<b>from</b>	<b>array</b>	From is the list of sources whose traffic this rule applies to. If any AdminNetworkPolicyIngressPeer matches the source of incoming traffic then the specified action is applied. This field must be defined and contain at least one item.  Support: Core
<b>from[]</b>	<b>object</b>	AdminNetworkPolicyIngressPeer defines an in-cluster peer to allow traffic from. Exactly one of the selector pointers must be set for a given peer. If a consumer observes none of its fields are set, they must assume an unknown option has been specified and fail closed.

Property	Type	Description
<b>name</b>	<b>string</b>	Name is an identifier for this rule, that may be no more than 100 characters in length. This field should be used by the implementation to help improve observability, readability and error-reporting for any applied BaselineAdminNetworkPolicies.  Support: Core
<b>ports</b>	<b>array</b>	Ports allows for matching traffic based on port and protocols. This field is a list of ports which should be matched on the pods selected for this policy i.e the subject of the policy. So it matches on the destination port for the ingress traffic. If Ports is not set then the rule does not filter traffic via port.  Support: Core
<b>ports[]</b>	<b>object</b>	AdminNetworkPolicyPort describes how to select network ports on pod(s). Exactly one field must be set.

#### 4.1.25. .spec.ingress[].from

##### Description

From is the list of sources whose traffic this rule applies to. If any AdminNetworkPolicyIngressPeer matches the source of incoming traffic then the specified action is applied. This field must be defined and contain at least one item.

Support: Core

##### Type

**array**

#### 4.1.26. .spec.ingress[].from[]

##### Description

AdminNetworkPolicyIngressPeer defines an in-cluster peer to allow traffic from. Exactly one of the selector pointers must be set for a given peer. If a consumer observes none of its fields are set, they must assume an unknown option has been specified and fail closed.

##### Type

**object**

Property	Type	Description
<b>namespaces</b>	<b>object</b>	Namespaces defines a way to select all pods within a set of Namespaces. Note that host-networked pods are not included in this type of peer.  Support: Core
<b>pods</b>	<b>object</b>	Pods defines a way to select a set of pods in a set of namespaces. Note that host-networked pods are not included in this type of peer.  Support: Core

**4.1.27. .spec.ingress[].from[].namespaces****Description**

Namespaces defines a way to select all pods within a set of Namespaces. Note that host-networked pods are not included in this type of peer.

Support: Core

**Type****object**

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

#### 4.1.28. .spec.ingress[].from[].namespaces.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.29. .spec.ingress[].from[].namespaces.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

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

#### 4.1.30. .spec.ingress[].from[].pods

##### Description

Pods defines a way to select a set of pods in a set of namespaces. Note that host-networked pods are not included in this type of peer.

Support: Core

##### Type

**object**

**Required**

- **namespaceSelector**
- **podSelector**

Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.
<b>podSelector</b>	<b>object</b>	PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

**4.1.31. .spec.ingress[].from[].pods.namespaceSelector****Description**

NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.

**Type**

**object**

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

**4.1.32. .spec.ingress[].from[].pods.namespaceSelector.matchExpressions****Description**

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

**Type****array****4.1.33. .spec.ingress[].from[].pods.namespaceSelector.matchExpressions[]****Description**

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

**Type****object****Required**

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

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

**4.1.34. .spec.ingress[].from[].pods.podSelector****Description**

PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

**Type****object**

Property	Type	Description
<b>matchExpressions</b>	<b>array</b>	matchExpressions is a list of label selector requirements. The requirements are ANDed.

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

#### 4.1.35. .spec.ingress[].from[].pods.podSelector.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.36. .spec.ingress[].from[].pods.podSelector.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

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

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

### 4.1.37. .spec.ingress[].ports

#### Description

Ports allows for matching traffic based on port and protocols. This field is a list of ports which should be matched on the pods selected for this policy i.e the subject of the policy. So it matches on the destination port for the ingress traffic. If Ports is not set then the rule does not filter traffic via port.

Support: Core

#### Type

**array**

### 4.1.38. .spec.ingress[].ports[]

#### Description

AdminNetworkPolicyPort describes how to select network ports on pod(s). Exactly one field must be set.

#### Type

**object**

Property	Type	Description
<b>namedPort</b>	<b>string</b>	NamedPort selects a port on a pod(s) based on name.  Support: Extended  <network-policy-api:experimental>
<b>portNumber</b>	<b>object</b>	Port selects a port on a pod(s) based on number.  Support: Core



Property	Type	Description
<b>portRange</b>	<b>object</b>	<p>PortRange selects a port range on a pod(s) based on provided start and end values.</p> <p>Support: Core</p>

#### 4.1.39. .spec.ingress[].ports[].portNumber

##### Description

Port selects a port on a pod(s) based on number.  
Support: Core

##### Type

**object**

##### Required

- **port**
- **protocol**

Property	Type	Description
<b>port</b>	<b>integer</b>	<p>Number defines a network port value.</p> <p>Support: Core</p>
<b>protocol</b>	<b>string</b>	<p>Protocol is the network protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.</p> <p>Support: Core</p>

#### 4.1.40. .spec.ingress[].ports[].portRange

##### Description

PortRange selects a port range on a pod(s) based on provided start and end values.  
Support: Core

##### Type

**object**

##### Required

- **end**

- **start**

Property	Type	Description
<b>end</b>	<b>integer</b>	End defines a network port that is the end of a port range, the End value must be greater than Start.  Support: Core
<b>protocol</b>	<b>string</b>	Protocol is the network protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.  Support: Core
<b>start</b>	<b>integer</b>	Start defines a network port that is the start of a port range, the Start value must be less than End.  Support: Core

#### 4.1.41. .spec.subject

##### Description

Subject defines the pods to which this BaselineAdminNetworkPolicy applies. Note that host-networked pods are not included in subject selection.

Support: Core

##### Type

**object**

Property	Type	Description
<b>namespaces</b>	<b>object</b>	Namespaces is used to select pods via namespace selectors.
<b>pods</b>	<b>object</b>	Pods is used to select pods via namespace AND pod selectors.

#### 4.1.42. .spec.subject.namespaces

##### Description

Namespaces is used to select pods via namespace selectors.

##### Type

**object**

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

#### 4.1.43. .spec.subject.namespaces.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.44. .spec.subject.namespaces.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

Property	Type	Description
<b>key</b>	<b>string</b>	key is the label key that the selector applies to.

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

#### 4.1.45. .spec.subject.pods

##### Description

Pods is used to select pods via namespace AND pod selectors.

##### Type

**object**

##### Required

- **namespaceSelector**
- **podSelector**

Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.
<b>podSelector</b>	<b>object</b>	PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

#### 4.1.46. .spec.subject.pods.namespaceSelector

##### Description

NamespaceSelector follows standard label selector semantics; if empty, it selects all Namespaces.

##### Type

**object**

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

#### 4.1.47. .spec.subject.pods.namespaceSelector.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.48. .spec.subject.pods.namespaceSelector.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

Property	Type	Description
<b>key</b>	<b>string</b>	key is the label key that the selector applies to.

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

#### 4.1.49. .spec.subject.pods.podSelector

##### Description

PodSelector is used to explicitly select pods within a namespace; if empty, it selects all Pods.

##### Type

**object**

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

#### 4.1.50. .spec.subject.pods.podSelector.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.51. .spec.subject.pods.podSelector.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

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

#### 4.1.52. .status

##### Description

Status is the status to be reported by the implementation.

##### Type

**object**

##### Required

- **conditions**

Property	Type	Description
<b>conditions</b>	<b>array</b>	

Property	Type	Description
<b>conditions[]</b>	<b>object</b>	<p>Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path <code>.status.conditions</code>. For example,</p> <pre> type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition <b>json:"conditions,omitempty"</b> <b>patchStrategy:"merge"</b> <b>patchMergeKey:"type"</b> <b>protobuf:"bytes,1,rep,name=conditions"</b>  // other fields }</pre>

#### 4.1.53. `.status.conditions`

Description

Type

**array**

#### 4.1.54. `.status.conditions[]`

Description

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

```

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

    // other fields
}
```



-

Type

object

Required

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

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

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

## 4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies**
  - **DELETE**: delete collection of BaselineAdminNetworkPolicy
  - **GET**: list objects of kind BaselineAdminNetworkPolicy
  - **POST**: create a BaselineAdminNetworkPolicy
- **/apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies/{name}**
  - **DELETE**: delete a BaselineAdminNetworkPolicy
  - **GET**: read the specified BaselineAdminNetworkPolicy
  - **PATCH**: partially update the specified BaselineAdminNetworkPolicy
  - **PUT**: replace the specified BaselineAdminNetworkPolicy
- **/apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies/{name}/status**
  - **GET**: read status of the specified BaselineAdminNetworkPolicy
  - **PATCH**: partially update status of the specified BaselineAdminNetworkPolicy
  - **PUT**: replace status of the specified BaselineAdminNetworkPolicy

### 4.2.1. /apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies

HTTP method

**DELETE**

Description

delete collection of BaselineAdminNetworkPolicy

Table 4.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

## HTTP method

**GET**

## Description

list objects of kind BaselineAdminNetworkPolicy

Table 4.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">BaselineAdminNetworkPolicyList</a> schema
401 - Unauthorized	Empty

## HTTP method

**POST**

## Description

create a BaselineAdminNetworkPolicy

Table 4.3. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 4.4. Body parameters

Parameter	Type	Description
<b>body</b>	<b>BaselineAdminNetworkPolicy</b> schema	

Table 4.5. HTTP responses

HTTP code	Response body
200 - OK	<b>BaselineAdminNetworkPolicy</b> schema
201 - Created	<b>BaselineAdminNetworkPolicy</b> schema
202 - Accepted	<b>BaselineAdminNetworkPolicy</b> schema
401 - Unauthorized	Empty

#### 4.2.2. /apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies/{name}

Table 4.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the BaselineAdminNetworkPolicy

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

delete a BaselineAdminNetworkPolicy

**Table 4.7. Query parameters**

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

**Table 4.8. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

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

read the specified BaselineAdminNetworkPolicy

**Table 4.9. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>BaselineAdminNetworkPolicy</b> schema
401 - Unauthorized	Empty

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

partially update the specified BaselineAdminNetworkPolicy

**Table 4.10. Query parameters**

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

Table 4.11. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">BaselineAdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

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

replace the specified BaselineAdminNetworkPolicy

Table 4.12. Query parameters

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

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

Table 4.13. Body parameters

Parameter	Type	Description
<b>body</b>	<b>BaselineAdminNetworkPolicy</b> schema	

Table 4.14. HTTP responses

HTTP code	Response body
200 - OK	<b>BaselineAdminNetworkPolicy</b> schema
201 - Created	<b>BaselineAdminNetworkPolicy</b> schema
401 - Unauthorized	Empty

#### 4.2.3. /apis/policy.networking.k8s.io/v1alpha1/baselineadminnetworkpolicies/{name}/

Table 4.15. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the BaselineAdminNetworkPolicy

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

read status of the specified BaselineAdminNetworkPolicy

**Table 4.16. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">BaselineAdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

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

partially update status of the specified BaselineAdminNetworkPolicy

**Table 4.17. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 4.18. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">BaselineAdminNetworkPolicy</a> schema
401 - Unauthorized	Empty

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

replace status of the specified BaselineAdminNetworkPolicy

Table 4.19. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 4.20. Body parameters

Parameter	Type	Description
<b>body</b>	<b>BaselineAdminNetworkPolicy</b> schema	

Table 4.21. HTTP responses

HTTP code	Response body
200 - OK	<b>BaselineAdminNetworkPolicy</b> schema
201 - Created	<b>BaselineAdminNetworkPolicy</b> schema
401 - Unauthorized	Empty

## CHAPTER 5. CLOUDPRIVATEIPCONFIG [CLOUD.NETWORK.OPENSIFT.IO/V1]

### Description

CloudPrivateIPConfig performs an assignment of a private IP address to the primary NIC associated with cloud VMs. This is done by specifying the IP and Kubernetes node which the IP should be assigned to. This CRD is intended to be used by the network plugin which manages the cluster network. The spec side represents the desired state requested by the network plugin, and the status side represents the current state that this CRD's controller has executed. No users will have permission to modify it, and if a cluster-admin decides to edit it for some reason, their changes will be overwritten the next time the network plugin reconciles the object. Note: the CR's name must specify the requested private IP address (can be IPv4 or IPv6).

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

### Type

**object**

### Required

- **spec**

## 5.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>

Property	Type	Description
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	spec is the definition of the desired private IP request.
<b>status</b>	<b>object</b>	status is the observed status of the desired private IP request. Read-only.

### 5.1.1. .spec

#### Description

spec is the definition of the desired private IP request.

#### Type

**object**

Property	Type	Description
<b>node</b>	<b>string</b>	node is the node name, as specified by the Kubernetes field: node.metadata.name

### 5.1.2. .status

#### Description

status is the observed status of the desired private IP request. Read-only.

#### Type

**object**

#### Required

- **conditions**

Property	Type	Description
<b>conditions</b>	<b>array</b>	condition is the assignment condition of the private IP and its status

Property	Type	Description
<b>conditions[]</b>	<b>object</b>	Condition contains details for one aspect of the current state of this API Resource.
<b>node</b>	<b>string</b>	node is the node name, as specified by the Kubernetes field: node.metadata.name

### 5.1.3. .status.conditions

#### Description

condition is the assignment condition of the private IP and its status

#### Type

**array**

### 5.1.4. .status.conditions[]

#### Description

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

#### Type

**object**

#### Required

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

Property	Type	Description
<b>lastTransitionTime</b>	<b>string</b>	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.

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

## 5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/cloud.network.openshift.io/v1/cloudprivateipconfigs**
  - **DELETE**: delete collection of CloudPrivateIPConfig
  - **GET**: list objects of kind CloudPrivateIPConfig
  - **POST**: create a CloudPrivateIPConfig
- **/apis/cloud.network.openshift.io/v1/cloudprivateipconfigs/{name}**
  - **DELETE**: delete a CloudPrivateIPConfig
  - **GET**: read the specified CloudPrivateIPConfig
  - **PATCH**: partially update the specified CloudPrivateIPConfig

- **PUT**: replace the specified CloudPrivateIPConfig
- **/apis/cloud.network.openshift.io/v1/cloudprivateipconfigs/{name}/status**
  - **GET**: read status of the specified CloudPrivateIPConfig
  - **PATCH**: partially update status of the specified CloudPrivateIPConfig
  - **PUT**: replace status of the specified CloudPrivateIPConfig

### 5.2.1. /apis/cloud.network.openshift.io/v1/cloudprivateipconfigs

#### HTTP method

#### DELETE

#### Description

delete collection of CloudPrivateIPConfig

Table 5.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

list objects of kind CloudPrivateIPConfig

Table 5.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">CloudPrivateIPConfigList</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### POST

#### Description

create a CloudPrivateIPConfig

Table 5.3. Query parameters

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

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

Table 5.4. Body parameters

Parameter	Type	Description
<b>body</b>	<b>CloudPrivateIPConfig</b> schema	

Table 5.5. HTTP responses

HTTP code	Response body
200 - OK	<b>CloudPrivateIPConfig</b> schema
201 - Created	<b>CloudPrivateIPConfig</b> schema
202 - Accepted	<b>CloudPrivateIPConfig</b> schema
401 - Unauthorized	Empty



## 5.2.2. /apis/cloud.network.openshift.io/v1/cloudprivateipconfigs/{name}

Table 5.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the CloudPrivateIPConfig

## HTTP method

**DELETE**

## Description

delete a CloudPrivateIPConfig

Table 5.7. Query parameters

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

Table 5.8. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

## HTTP method

**GET**

## Description

read the specified CloudPrivateIPConfig

Table 5.9. HTTP responses

HTTP code	Reponse body
200 - OK	<b>CloudPrivateIPConfig</b> schema
401 - Unauthorized	Empty

## HTTP method

**PATCH****Description**

partially update the specified CloudPrivateIPConfig

**Table 5.10. Query parameters**

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

**Table 5.11. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">CloudPrivateIPConfig</a> schema
401 - Unauthorized	Empty

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

replace the specified CloudPrivateIPConfig

**Table 5.12. Query parameters**

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

Table 5.13. Body parameters

Parameter	Type	Description
<b>body</b>	<b>CloudPrivateIPConfig</b> schema	

Table 5.14. HTTP responses

HTTP code	Response body
200 - OK	<b>CloudPrivateIPConfig</b> schema
201 - Created	<b>CloudPrivateIPConfig</b> schema
401 - Unauthorized	Empty

### 5.2.3. /apis/cloud.network.openshift.io/v1/cloudprivateipconfigs/{name}/status

Table 5.15. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the CloudPrivateIPConfig

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

read status of the specified CloudPrivateIPConfig

**Table 5.16. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">CloudPrivateIPConfig</a> schema
401 - Unauthorized	Empty

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

partially update status of the specified CloudPrivateIPConfig

**Table 5.17. Query parameters**

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

Table 5.18. HTTP responses

HTTP code	Response body
200 - OK	<b>CloudPrivateIPConfig</b> schema
401 - Unauthorized	Empty

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

replace status of the specified CloudPrivateIPConfig

Table 5.19. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 5.20. Body parameters

Parameter	Type	Description
<b>body</b>	<b>CloudPrivateIPConfig</b> schema	

Table 5.21. HTTP responses

HTTP code	Response body
200 - OK	<b>CloudPrivateIPConfig</b> schema
201 - Created	<b>CloudPrivateIPConfig</b> schema
401 - Unauthorized	Empty

## CHAPTER 6. EGRESSFIREWALL [K8S.OVN.ORG/V1]

### Description

EgressFirewall describes the current egress firewall for a Namespace. Traffic from a pod to an IP address outside the cluster will be checked against each EgressFirewallRule in the pod's namespace's EgressFirewall, in order. If no rule matches (or no EgressFirewall is present) then the traffic will be allowed by default.

### Type

**object**

### Required

- **spec**

## 6.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	Specification of the desired behavior of EgressFirewall.

Property	Type	Description
<b>status</b>	<b>object</b>	Observed status of EgressFirewall

### 6.1.1. .spec

#### Description

Specification of the desired behavior of EgressFirewall.

#### Type

**object**

#### Required

- **egress**

Property	Type	Description
<b>egress</b>	<b>array</b>	a collection of egress firewall rule objects
<b>egress[]</b>	<b>object</b>	EgressFirewallRule is a single egressfirewall rule object

### 6.1.2. .spec.egress

#### Description

a collection of egress firewall rule objects

#### Type

**array**

### 6.1.3. .spec.egress[]

#### Description

EgressFirewallRule is a single egressfirewall rule object

#### Type

**object**

#### Required

- **to**
- **type**

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



Property	Type	Description
<b>ports</b>	<b>array</b>	ports specify what ports and protocols the rule applies to
<b>ports[]</b>	<b>object</b>	EgressFirewallPort specifies the port to allow or deny traffic to
<b>to</b>	<b>object</b>	to is the target that traffic is allowed/denied to
<b>type</b>	<b>string</b>	type marks this as an "Allow" or "Deny" rule

#### 6.1.4. .spec.egress[].ports

##### Description

ports specify what ports and protocols the rule applies to

##### Type

**array**

#### 6.1.5. .spec.egress[].ports[]

##### Description

EgressFirewallPort specifies the port to allow or deny traffic to

##### Type

**object**

##### Required

- **port**
- **protocol**

Property	Type	Description
<b>port</b>	<b>integer</b>	port that the traffic must match
<b>protocol</b>	<b>string</b>	protocol (tcp, udp, sctp) that the traffic must match.

#### 6.1.6. .spec.egress[].to

##### Description

to is the target that traffic is allowed/denied to

##### Type

**object**

Property	Type	Description
<b>cidrSelector</b>	<b>string</b>	cidrSelector is the CIDR range to allow/deny traffic to. If this is set, dnsName and nodeSelector must be unset.
<b>dnsName</b>	<b>string</b>	dnsName is the domain name to allow/deny traffic to. If this is set, cidrSelector and nodeSelector must be unset. For a wildcard DNS name, the " <b>will match only one label. Additionally, only a single</b> " can be used at the beginning of the wildcard DNS name. For example, '*.example.com' will match 'sub1.example.com' but won't match 'sub2.sub1.example.com'.
<b>nodeSelector</b>	<b>object</b>	nodeSelector will allow/deny traffic to the Kubernetes node IP of selected nodes. If this is set, cidrSelector and DNSName must be unset.

### 6.1.7. .spec.egress[].to.nodeSelector

#### Description

nodeSelector will allow/deny traffic to the Kubernetes node IP of selected nodes. If this is set, cidrSelector and DNSName must be unset.

#### Type

**object**

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

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

### 6.1.8. .spec.egress[].to.nodeSelector.matchExpressions

#### Description

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

#### Type

**array**

### 6.1.9. .spec.egress[].to.nodeSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

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

### 6.1.10. .status

#### Description

Observed status of EgressFirewall

#### Type

**object**

Property	Type	Description
<b>messages</b>	<b>array (string)</b>	
<b>status</b>	<b>string</b>	

## 6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/k8s.ovn.org/v1/egressfirewalls**
  - **GET**: list objects of kind EgressFirewall
- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls**
  - **DELETE**: delete collection of EgressFirewall
  - **GET**: list objects of kind EgressFirewall
  - **POST**: create an EgressFirewall
- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls/{name}**
  - **DELETE**: delete an EgressFirewall
  - **GET**: read the specified EgressFirewall
  - **PATCH**: partially update the specified EgressFirewall
  - **PUT**: replace the specified EgressFirewall

- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls/{name}/status**
  - **GET**: read status of the specified EgressFirewall
  - **PATCH**: partially update status of the specified EgressFirewall
  - **PUT**: replace status of the specified EgressFirewall

### 6.2.1. /apis/k8s.ovn.org/v1/egressfirewalls

#### HTTP method

#### GET

#### Description

list objects of kind EgressFirewall

Table 6.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">EgressFirewallList</a> schema
401 - Unauthorized	Empty

### 6.2.2. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls

#### HTTP method

#### DELETE

#### Description

delete collection of EgressFirewall

Table 6.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

list objects of kind EgressFirewall

Table 6.3. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">EgressFirewallList</a> schema

HTTP code	Response body
401 - Unauthorized	Empty

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

create an EgressFirewall

**Table 6.4. Query parameters**

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

**Table 6.5. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>EgressFirewall</b> schema	

**Table 6.6. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">EgressFirewall</a> schema
201 - Created	<a href="#">EgressFirewall</a> schema
202 - Accepted	<a href="#">EgressFirewall</a> schema
401 - Unauthorized	Empty

### 6.2.3. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls/{name}

Table 6.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressFirewall

#### HTTP method

#### DELETE

#### Description

delete an EgressFirewall

Table 6.8. Query parameters

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

Table 6.9. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

read the specified EgressFirewall

**Table 6.10. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EgressFirewall</a> schema
401 - Unauthorized	Empty

## HTTP method

### PATCH

## Description

partially update the specified EgressFirewall

**Table 6.11. Query parameters**

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

**Table 6.12. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EgressFirewall</a> schema



HTTP code	Response body
401 - Unauthorized	Empty

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

replace the specified EgressFirewall

**Table 6.13. Query parameters**

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

**Table 6.14. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>EgressFirewall</b> schema	

**Table 6.15. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">EgressFirewall</a> schema
201 - Created	<a href="#">EgressFirewall</a> schema
401 - Unauthorized	Empty

#### 6.2.4. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressfirewalls/{name}/status

Table 6.16. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressFirewall

#### HTTP method

##### GET

#### Description

read status of the specified EgressFirewall

Table 6.17. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">EgressFirewall</a> schema
401 - Unauthorized	Empty

#### HTTP method

##### PATCH

#### Description

partially update status of the specified EgressFirewall

Table 6.18. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 6.19. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressFirewall</b> schema
401 - Unauthorized	Empty

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

replace status of the specified EgressFirewall

Table 6.20. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 6.21. Body parameters

Parameter	Type	Description
<b>body</b>	<b>EgressFirewall</b> schema	

Table 6.22. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressFirewall</b> schema
201 - Created	<b>EgressFirewall</b> schema
401 - Unauthorized	Empty

## CHAPTER 7. EGRESSIP [K8S.OVN.ORG/V1]

### Description

EgressIP is a CRD allowing the user to define a fixed source IP for all egress traffic originating from any pods which match the EgressIP resource according to its spec definition.

### Type

**object**

### Required

- **spec**

## 7.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	Specification of the desired behavior of EgressIP.
<b>status</b>	<b>object</b>	Observed status of EgressIP. Read-only.

### 7.1.1. .spec

#### Description

Specification of the desired behavior of EgressIP.

#### Type

**object**

#### Required

- **egressIPs**
- **namespaceSelector**

Property	Type	Description
<b>egressIPs</b>	<b>array (string)</b>	EgressIPs is the list of egress IP addresses requested. Can be IPv4 and/or IPv6. This field is mandatory.
<b>namespaceSelector</b>	<b>object</b>	NamespaceSelector applies the egress IP only to the namespace(s) whose label matches this definition. This field is mandatory.
<b>podSelector</b>	<b>object</b>	PodSelector applies the egress IP only to the pods whose label matches this definition. This field is optional, and in case it is not set: results in the egress IP being applied to all pods in the namespace(s) matched by the NamespaceSelector. In case it is set: is intersected with the NamespaceSelector, thus applying the egress IP to the pods (in the namespace(s) already matched by the NamespaceSelector) which match this pod selector.

### 7.1.2. .spec.namespaceSelector

#### Description

NamespaceSelector applies the egress IP only to the namespace(s) whose label matches this definition. This field is mandatory.

#### Type

**object**

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

### 7.1.3. .spec.namespaceSelector.matchExpressions

#### Description

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

#### Type

**array**

### 7.1.4. .spec.namespaceSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

Property	Type	Description
<b>key</b>	<b>string</b>	key is the label key that the selector applies to.

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

### 7.1.5. .spec.podSelector

#### Description

PodSelector applies the egress IP only to the pods whose label matches this definition. This field is optional, and in case it is not set: results in the egress IP being applied to all pods in the namespace(s) matched by the NamespaceSelector. In case it is set: is intersected with the NamespaceSelector, thus applying the egress IP to the pods (in the namespace(s) already matched by the NamespaceSelector) which match this pod selector.

#### Type

**object**

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

### 7.1.6. .spec.podSelector.matchExpressions



**Description**

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

**Type**

**array**

**7.1.7. .spec.podSelector.matchExpressions[]****Description**

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

**Type**

**object**

**Required**

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

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

**7.1.8. .status****Description**

Observed status of EgressIP. Read-only.

**Type**

**object**

**Required**

- **items**

Property	Type	Description
<b>items</b>	<b>array</b>	The list of assigned egress IPs and their corresponding node assignment.
<b>items[]</b>	<b>object</b>	The per node status, for those egress IPs who have been assigned.

### 7.1.9. .status.items

#### Description

The list of assigned egress IPs and their corresponding node assignment.

#### Type

**array**

### 7.1.10. .status.items[]

#### Description

The per node status, for those egress IPs who have been assigned.

#### Type

**object**

#### Required

- **egressIP**
- **node**

Property	Type	Description
<b>egressIP</b>	<b>string</b>	Assigned egress IP
<b>node</b>	<b>string</b>	Assigned node name

## 7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/k8s.ovn.org/v1/egressips**
  - **DELETE**: delete collection of EgressIP
  - **GET**: list objects of kind EgressIP
  - **POST**: create an EgressIP
- **/apis/k8s.ovn.org/v1/egressips/{name}**

- **DELETE**: delete an EgressIP
- **GET**: read the specified EgressIP
- **PATCH**: partially update the specified EgressIP
- **PUT**: replace the specified EgressIP

### 7.2.1. /apis/k8s.ovn.org/v1/egressips

#### HTTP method

##### **DELETE**

#### Description

delete collection of EgressIP

Table 7.1. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
401 - Unauthorized	Empty

#### HTTP method

##### **GET**

#### Description

list objects of kind EgressIP

Table 7.2. HTTP responses

HTTP code	Reponse body
200 - OK	<b>EgressIPList</b> schema
401 - Unauthorized	Empty

#### HTTP method

##### **POST**

#### Description

create an EgressIP

Table 7.3. Query parameters

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

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

Table 7.4. Body parameters

Parameter	Type	Description
<b>body</b>	<b>EgressIP</b> schema	

Table 7.5. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressIP</b> schema
201 - Created	<b>EgressIP</b> schema
202 - Accepted	<b>EgressIP</b> schema
401 - Unauthorized	Empty

### 7.2.2. /apis/k8s.ovn.org/v1/egressips/{name}

Table 7.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressIP

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

delete an EgressIP

**Table 7.7. Query parameters**

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

**Table 7.8. HTTP responses**

HTTP code	Response body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

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

read the specified EgressIP

**Table 7.9. HTTP responses**

HTTP code	Response body
200 - OK	<b>EgressIP</b> schema
401 - Unauthorized	Empty

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

partially update the specified EgressIP

Table 7.10. Query parameters

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

Table 7.11. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressIP</b> schema
401 - Unauthorized	Empty

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

replace the specified EgressIP

Table 7.12. Query parameters

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

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

Table 7.13. Body parameters

Parameter	Type	Description
<b>body</b>	<b>EgressIP</b> schema	

Table 7.14. HTTP responses

HTTP code	Reponse body
200 - OK	<b>EgressIP</b> schema
201 - Created	<b>EgressIP</b> schema
401 - Unauthorized	Empty

## CHAPTER 8. EGRESSQOS [K8S.OVN.ORG/V1]

### Description

EgressQoS is a CRD that allows the user to define a DSCP value for pods egress traffic on its namespace to specified CIDRs. Traffic from these pods will be checked against each EgressQoSRule in the namespace's EgressQoS, and if there is a match the traffic is marked with the relevant DSCP value.

### Type

**object**

## 8.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	EgressQoSSpec defines the desired state of EgressQoS
<b>status</b>	<b>object</b>	EgressQoSStatus defines the observed state of EgressQoS



### 8.1.1. .spec

#### Description

EgressQoSSpec defines the desired state of EgressQoS

#### Type

**object**

#### Required

- **egress**

Property	Type	Description
<b>egress</b>	<b>array</b>	a collection of Egress QoS rule objects
<b>egress[]</b>	<b>object</b>	

### 8.1.2. .spec.egress

#### Description

a collection of Egress QoS rule objects

#### Type

**array**

### 8.1.3. .spec.egress[]

#### Description

#### Type

**object**

#### Required

- **dscp**

Property	Type	Description
<b>dscp</b>	<b>integer</b>	DSCP marking value for matching pods' traffic.
<b>dstCIDR</b>	<b>string</b>	DstCIDR specifies the destination's CIDR. Only traffic heading to this CIDR will be marked with the DSCP value. This field is optional, and in case it is not set the rule is applied to all egress traffic regardless of the destination.

Property	Type	Description
<b>podSelector</b>	<b>object</b>	PodSelector applies the QoS rule only to the pods in the namespace whose label matches this definition. This field is optional, and in case it is not set results in the rule being applied to all pods in the namespace.

#### 8.1.4. .spec.egress[].podSelector

##### Description

PodSelector applies the QoS rule only to the pods in the namespace whose label matches this definition. This field is optional, and in case it is not set results in the rule being applied to all pods in the namespace.

##### Type

**object**

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

#### 8.1.5. .spec.egress[].podSelector.matchExpressions

##### Description

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

##### Type

**array**

### 8.1.6. .spec.egress[].podSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

### 8.1.7. .status

#### Description

EgressQoSStatus defines the observed state of EgressQoS

#### Type

**object**

Property	Type	Description
<b>conditions</b>	<b>array</b>	An array of condition objects indicating details about status of EgressQoS object.

Property	Type	Description
<b>conditions[]</b>	<b>object</b>	<p>Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path <code>.status.conditions</code>. For example,</p> <pre> type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition <b>json:"conditions,omitempty"</b> <b>patchStrategy:"merge"</b> <b>patchMergeKey:"type"</b> <b>protobuf:"bytes,1,rep,name=conditions"</b>  // other fields }</pre>
<b>status</b>	<b>string</b>	A concise indication of whether the EgressQoS resource is applied with success.

### 8.1.8. `.status.conditions`

#### Description

An array of condition objects indicating details about status of EgressQoS object.

#### Type

**array**

### 8.1.9. `.status.conditions[]`

#### Description

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

```

type FooStatus struct{
// Represents the observations of a foo's current state.
// Known .status.conditions.type are: "Available", "Progressing", and "Degraded"
// +patchMergeKey=type
// +patchStrategy=merge
```

```

// +listType=map
// +listMapKey=type
Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge"
patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"

// other fields
}

```

Type

**object**

Required

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

Property	Type	Description
<b>lastTransitionTime</b>	<b>string</b>	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
<b>message</b>	<b>string</b>	message is a human readable message indicating details about the transition. This may be an empty string.
<b>observedGeneration</b>	<b>integer</b>	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.

Property	Type	Description
<b>reason</b>	<b>string</b>	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
<b>status</b>	<b>string</b>	status of the condition, one of True, False, Unknown.
<b>type</b>	<b>string</b>	type of condition in CamelCase or in foo.example.com/CamelCase. -- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

## 8.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/k8s.ovn.org/v1/egressqoses**
  - **GET**: list objects of kind EgressQoS
- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses**
  - **DELETE**: delete collection of EgressQoS
  - **GET**: list objects of kind EgressQoS
  - **POST**: create an EgressQoS
- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses/{name}**
  - **DELETE**: delete an EgressQoS
  - **GET**: read the specified EgressQoS
  - **PATCH**: partially update the specified EgressQoS
  - **PUT**: replace the specified EgressQoS

- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses/{name}/status**
  - **GET**: read status of the specified EgressQoS
  - **PATCH**: partially update status of the specified EgressQoS
  - **PUT**: replace status of the specified EgressQoS

### 8.2.1. /apis/k8s.ovn.org/v1/egressqoses

#### HTTP method

#### GET

#### Description

list objects of kind EgressQoS

Table 8.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">EgressQoSList</a> schema
401 - Unauthorized	Empty

### 8.2.2. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses

#### HTTP method

#### DELETE

#### Description

delete collection of EgressQoS

Table 8.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

list objects of kind EgressQoS

Table 8.3. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">EgressQoSList</a> schema

HTTP code	Response body
401 - Unauthorized	Empty

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

create an EgressQoS

**Table 8.4. Query parameters**

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

**Table 8.5. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>EgressQoS</b> schema	

**Table 8.6. HTTP responses**



HTTP code	Reponse body
200 - OK	<a href="#">EgressQoS</a> schema
201 - Created	<a href="#">EgressQoS</a> schema
202 - Accepted	<a href="#">EgressQoS</a> schema
401 - Unauthorized	Empty

### 8.2.3. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses/{name}

Table 8.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressQoS

#### HTTP method

#### DELETE

#### Description

delete an EgressQoS

Table 8.8. Query parameters

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

Table 8.9. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

**Description**

read the specified EgressQoS

**Table 8.10. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EgressQoS</a> schema
401 - Unauthorized	Empty

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

partially update the specified EgressQoS

**Table 8.11. Query parameters**

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

**Table 8.12. HTTP responses**

HTTP code	Response body
200 - OK	<b>EgressQoS</b> schema
401 - Unauthorized	Empty

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

replace the specified EgressQoS

**Table 8.13. Query parameters**

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

**Table 8.14. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>EgressQoS</b> schema	

**Table 8.15. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">EgressQoS</a> schema
201 - Created	<a href="#">EgressQoS</a> schema
401 - Unauthorized	Empty

#### 8.2.4. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressqoses/{name}/status

Table 8.16. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressQoS

#### HTTP method

##### GET

#### Description

read status of the specified EgressQoS

Table 8.17. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">EgressQoS</a> schema
401 - Unauthorized	Empty

#### HTTP method

##### PATCH

#### Description

partially update status of the specified EgressQoS

Table 8.18. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 8.19. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressQoS</b> schema
401 - Unauthorized	Empty

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

replace status of the specified EgressQoS

Table 8.20. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 8.21. Body parameters

Parameter	Type	Description
<b>body</b>	<b>EgressQoS</b> schema	

Table 8.22. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressQoS</b> schema
201 - Created	<b>EgressQoS</b> schema
401 - Unauthorized	Empty

## CHAPTER 9. EGRESSSERVICE [K8S.OVN.ORG/V1]

### Description

EgressService is a CRD that allows the user to request that the source IP of egress packets originating from all of the pods that are endpoints of the corresponding LoadBalancer Service would be its ingress IP. In addition, it allows the user to request that egress packets originating from all of the pods that are endpoints of the LoadBalancer service would use a different network than the main one.

### Type

**object**

## 9.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	EgressServiceSpec defines the desired state of EgressService
<b>status</b>	<b>object</b>	EgressServiceStatus defines the observed state of EgressService

### 9.1.1. .spec

#### Description

EgressServiceSpec defines the desired state of EgressService

#### Type

**object**

Property	Type	Description
<b>network</b>	<b>string</b>	The network which this service should send egress and corresponding ingress replies to. This is typically implemented as VRF mapping, representing a numeric id or string name of a routing table which by omission uses the default host routing.
<b>nodeSelector</b>	<b>object</b>	Allows limiting the nodes that can be selected to handle the service's traffic when sourceIPBy=LoadBalancerIP. When present only a node whose labels match the specified selectors can be selected for handling the service's traffic. When it is not specified any node in the cluster can be chosen to manage the service's traffic.
<b>sourceIPBy</b>	<b>string</b>	Determines the source IP of egress traffic originating from the pods backing the LoadBalancer Service. When <b>LoadBalancerIP</b> the source IP is set to its LoadBalancer ingress IP. When <b>Network</b> the source IP is set according to the interface of the Network, leveraging the masquerade rules that are already in place. Typically these rules specify SNAT to the IP of the outgoing interface, which means the packet will typically leave with the IP of the node.

### 9.1.2. .spec.nodeSelector

#### Description

Allows limiting the nodes that can be selected to handle the service's traffic when sourceIPBy=LoadBalancerIP. When present only a node whose labels match the specified selectors can be selected for handling the service's traffic. When it is not specified any node in the cluster can



be chosen to manage the service's traffic.

#### Type

**object**

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

### 9.1.3. .spec.nodeSelector.matchExpressions

#### Description

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

#### Type

**array**

### 9.1.4. .spec.nodeSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

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

### 9.1.5. .status

#### Description

EgressServiceStatus defines the observed state of EgressService

#### Type

**object**

#### Required

- **host**

Property	Type	Description
<b>host</b>	<b>string</b>	The name of the node selected to handle the service's traffic. In case sourceIPBy=Network the field will be set to "ALL".

## 9.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/k8s.ovn.org/v1/egressservices**
  - **GET**: list objects of kind EgressService
- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices**
  - **DELETE**: delete collection of EgressService
  - **GET**: list objects of kind EgressService

- **POST**: create an EgressService
- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices/{name}**
  - **DELETE**: delete an EgressService
  - **GET**: read the specified EgressService
  - **PATCH**: partially update the specified EgressService
  - **PUT**: replace the specified EgressService
- **/apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices/{name}/status**
  - **GET**: read status of the specified EgressService
  - **PATCH**: partially update status of the specified EgressService
  - **PUT**: replace status of the specified EgressService

### 9.2.1. /apis/k8s.ovn.org/v1/egressservices

HTTP method

**GET**

Description

list objects of kind EgressService

Table 9.1. HTTP responses

HTTP code	Reponse body
200 - OK	<b>EgressServiceList</b> schema
401 - Unauthorized	Empty

### 9.2.2. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices

HTTP method

**DELETE**

Description

delete collection of EgressService

Table 9.2. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
401 - Unauthorized	Empty

HTTP method

**GET****Description**

list objects of kind EgressService

**Table 9.3. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EgressServiceList</a> schema
401 - Unauthorized	Empty

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

create an EgressService

**Table 9.4. Query parameters**

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

**Table 9.5. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>EgressService</b> schema	

Table 9.6. HTTP responses

HTTP code	Reponse body
200 - OK	<b>EgressService</b> schema
201 - Created	<b>EgressService</b> schema
202 - Accepted	<b>EgressService</b> schema
401 - Unauthorized	Empty

### 9.2.3. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices/{name}

Table 9.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressService

#### HTTP method

#### DELETE

#### Description

delete an EgressService

Table 9.8. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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	<b>Status</b> schema

HTTP code	Reponse body
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

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

read the specified EgressService

**Table 9.10. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">EgressService</a> schema
401 - Unauthorized	Empty

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

partially update the specified EgressService

**Table 9.11. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 9.12. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressService</b> schema
401 - Unauthorized	Empty

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

replace the specified EgressService

Table 9.13. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 9.14. Body parameters

Parameter	Type	Description
<b>body</b>	<b>EgressService</b> schema	

Table 9.15. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressService</b> schema
201 - Created	<b>EgressService</b> schema
401 - Unauthorized	Empty

### 9.2.4. /apis/k8s.ovn.org/v1/namespaces/{namespace}/egressservices/{name}/status

Table 9.16. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressService

HTTP method



**GET****Description**

read status of the specified EgressService

**Table 9.17. HTTP responses**

HTTP code	Response body
200 - OK	<b>EgressService</b> schema
401 - Unauthorized	Empty

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

partially update status of the specified EgressService

**Table 9.18. Query parameters**

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

**Table 9.19. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EgressService</a> schema
401 - Unauthorized	Empty

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

replace status of the specified EgressService

**Table 9.20. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<b>fieldValidation</b>	<b>string</b>	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 9.21. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">EgressService</a> schema	

**Table 9.22. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>EgressService</b> schema
201 - Created	<b>EgressService</b> schema
401 - Unauthorized	Empty

## CHAPTER 10. ENDPOINTS [V1]

### Description

Endpoints is a collection of endpoints that implement the actual service. Example:

```
Name: "mysvc",
Subsets: [
  {
    Addresses: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}],
    Ports: [{"name": "a", "port": 8675}, {"name": "b", "port": 309}]
  },
  {
    Addresses: [{"ip": "10.10.3.3"}],
    Ports: [{"name": "a", "port": 93}, {"name": "b", "port": 76}]
  },
]
```

### Type

**object**

### 10.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>

Property	Type	Description
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>subsets</b>	<b>array</b>	The set of all endpoints is the union of all subsets. Addresses are placed into subsets according to the IPs they share. A single address with multiple ports, some of which are ready and some of which are not (because they come from different containers) will result in the address being displayed in different subsets for the different ports. No address will appear in both Addresses and NotReadyAddresses in the same subset. Sets of addresses and ports that comprise a service.
<b>subsets[]</b>	<b>object</b>	<p>EndpointSubset is a group of addresses with a common set of ports. The expanded set of endpoints is the Cartesian product of Addresses x Ports. For example, given:</p> <pre>{ Addresses: [{ "ip": "10.10.1.1"}, { "ip": "10.10.2.2"}], Ports: [{ "name": "a", "port": 8675}, { "name": "b", "port": 309}] }</pre> <p>The resulting set of endpoints can be viewed as:</p> <pre>a: [ 10.10.1.1:8675, 10.10.2.2:8675 ], b: [ 10.10.1.1:309, 10.10.2.2:309 ]</pre>

### 10.1.1. .subsets

#### Description

The set of all endpoints is the union of all subsets. Addresses are placed into subsets according to the IPs they share. A single address with multiple ports, some of which are ready and some of which are not (because they come from different containers) will result in the address being displayed in different subsets for the different ports. No address will appear in both Addresses and NotReadyAddresses in the same subset. Sets of addresses and ports that comprise a service.

#### Type

**array**

## 10.1.2. .subsets[]

### Description

EndpointSubset is a group of addresses with a common set of ports. The expanded set of endpoints is the Cartesian product of Addresses x Ports. For example, given:

```
{
  Addresses: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}],
  Ports:    [{"name": "a", "port": 8675}, {"name": "b", "port": 309}]
}
```

The resulting set of endpoints can be viewed as:

```
a: [ 10.10.1.1:8675, 10.10.2.2:8675 ],
b: [ 10.10.1.1:309, 10.10.2.2:309 ]
```

### Type

**object**

Property	Type	Description
<b>addresses</b>	<b>array</b>	IP addresses which offer the related ports that are marked as ready. These endpoints should be considered safe for load balancers and clients to utilize.
<b>addresses[]</b>	<b>object</b>	EndpointAddress is a tuple that describes single IP address.
<b>notReadyAddresses</b>	<b>array</b>	IP addresses which offer the related ports but are not currently marked as ready because they have not yet finished starting, have recently failed a readiness check, or have recently failed a liveness check.
<b>notReadyAddresses[]</b>	<b>object</b>	EndpointAddress is a tuple that describes single IP address.
<b>ports</b>	<b>array</b>	Port numbers available on the related IP addresses.
<b>ports[]</b>	<b>object</b>	EndpointPort is a tuple that describes a single port.

## 10.1.3. .subsets[].addresses

### Description

IP addresses which offer the related ports that are marked as ready. These endpoints should be considered safe for load balancers and clients to utilize.

#### Type

**array**

### 10.1.4. .subsets[].addresses[]

#### Description

EndpointAddress is a tuple that describes single IP address.

#### Type

**object**

#### Required

- **ip**

Property	Type	Description
<b>hostname</b>	<b>string</b>	The Hostname of this endpoint
<b>ip</b>	<b>string</b>	The IP of this endpoint. May not be loopback (127.0.0.0/8 or ::1), link-local (169.254.0.0/16 or fe80::/10), or link-local multicast (224.0.0.0/24 or ff02::/16).
<b>nodeName</b>	<b>string</b>	Optional: Node hosting this endpoint. This can be used to determine endpoints local to a node.
<b>targetRef</b>	<b>object</b>	ObjectReference contains enough information to let you inspect or modify the referred object.

### 10.1.5. .subsets[].addresses[].targetRef

#### Description

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

#### Type

**object**

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	API version of the referent.

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



Property	Type	Description
<b>uid</b>	<b>string</b>	UID of the referent. More info: <a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids</a>

### 10.1.6. .subsets[].notReadyAddresses

#### Description

IP addresses which offer the related ports but are not currently marked as ready because they have not yet finished starting, have recently failed a readiness check, or have recently failed a liveness check.

#### Type

**array**

### 10.1.7. .subsets[].notReadyAddresses[]

#### Description

EndpointAddress is a tuple that describes single IP address.

#### Type

**object**

#### Required

- **ip**

Property	Type	Description
<b>hostname</b>	<b>string</b>	The Hostname of this endpoint
<b>ip</b>	<b>string</b>	The IP of this endpoint. May not be loopback (127.0.0.0/8 or ::1), link-local (169.254.0.0/16 or fe80::/10), or link-local multicast (224.0.0.0/24 or ff02::/16).
<b>nodeName</b>	<b>string</b>	Optional: Node hosting this endpoint. This can be used to determine endpoints local to a node.
<b>targetRef</b>	<b>object</b>	ObjectReference contains enough information to let you inspect or modify the referred object.

### 10.1.8. .subsets[].notReadyAddresses[].targetRef

## Description

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

## Type

**object**

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

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

### 10.1.9. .subsets[].ports

#### Description

Port numbers available on the related IP addresses.

#### Type

**array**

### 10.1.10. .subsets[].ports[]

#### Description

EndpointPort is a tuple that describes a single port.

#### Type

**object**

#### Required

- **port**

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

Property	Type	Description
<b>appProtocol</b>	<b>string</b>	<p>The application protocol for this port. This is used as a hint for implementations to offer richer behavior for protocols that they understand. This field follows standard Kubernetes label syntax. Valid values are either:</p> <ul style="list-style-type: none"> <li>* Un-prefixed protocol names - reserved for IANA standard service names (as per RFC-6335 and <a href="https://www.iana.org/assignments/service-names">https://www.iana.org/assignments/service-names</a>).</li> <li>* Kubernetes-defined prefixed names: <ul style="list-style-type: none"> <li>* 'kubernetes.io/h2c' - HTTP/2 prior knowledge over cleartext as described in <a href="https://www.rfc-editor.org/rfc/rfc9113.html#name-starting-http-2-with-prior-">https://www.rfc-editor.org/rfc/rfc9113.html#name-starting-http-2-with-prior-</a></li> <li>* 'kubernetes.io/ws' - WebSocket over cleartext as described in <a href="https://www.rfc-editor.org/rfc/rfc6455">https://www.rfc-editor.org/rfc/rfc6455</a></li> <li>* 'kubernetes.io/wss' - WebSocket over TLS as described in <a href="https://www.rfc-editor.org/rfc/rfc6455">https://www.rfc-editor.org/rfc/rfc6455</a></li> </ul> </li> <li>* Other protocols should use implementation-defined prefixed names such as mycompany.com/my-custom-protocol.</li> </ul>
<b>name</b>	<b>string</b>	The name of this port. This must match the 'name' field in the corresponding ServicePort. Must be a DNS_LABEL. Optional only if one port is defined.
<b>port</b>	<b>integer</b>	The port number of the endpoint.

Property	Type	Description
<b>protocol</b>	<b>string</b>	<p>The IP protocol for this port. Must be UDP, TCP, or SCTP. Default is TCP.</p> <p>Possible enum values: - <b>"SCTP"</b> is the SCTP protocol. - <b>"TCP"</b> is the TCP protocol. - <b>"UDP"</b> is the UDP protocol.</p>

## 10.2. API ENDPOINTS

The following API endpoints are available:

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

### 10.2.1. /api/v1/endpoints

HTTP method

**GET****Description**

list or watch objects of kind Endpoints

**Table 10.1. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EndpointsList</a> schema
401 - Unauthorized	Empty

**10.2.2. /api/v1/watch/endpoints****HTTP method****GET****Description**

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

**Table 10.2. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

**10.2.3. /api/v1/namespaces/{namespace}/endpoints****HTTP method****DELETE****Description**

delete collection of Endpoints

**Table 10.3. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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	<a href="#">Status</a> schema
401 - Unauthorized	Empty

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

list or watch objects of kind Endpoints

**Table 10.5. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">EndpointsList</a> schema
401 - Unauthorized	Empty

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

create Endpoints

**Table 10.6. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 10.7. Body parameters

Parameter	Type	Description
<b>body</b>	<b>Endpoints</b> schema	

Table 10.8. HTTP responses

HTTP code	Response body
200 - OK	<b>Endpoints</b> schema
201 - Created	<b>Endpoints</b> schema
202 - Accepted	<b>Endpoints</b> schema
401 - Unauthorized	Empty

#### 10.2.4. /api/v1/watch/namespaces/{namespace}/endpoints

##### HTTP method

##### GET

##### Description

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

Table 10.9. HTTP responses



HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

### 10.2.5. /api/v1/namespaces/{namespace}/endpoints/{name}

Table 10.10. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Endpoints

#### HTTP method

#### DELETE

#### Description

delete Endpoints

Table 10.11. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

read the specified Endpoints

Table 10.13. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">Endpoints</a> schema
401 - Unauthorized	Empty

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

partially update the specified Endpoints

**Table 10.14. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<b>fieldValidation</b>	<b>string</b>	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	Response body
200 - OK	<a href="#">Endpoints</a> schema
201 - Created	<a href="#">Endpoints</a> schema

HTTP code	Response body
401 - Unauthorized	Empty

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

replace the specified Endpoints

**Table 10.16. Query parameters**

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

**Table 10.18. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Endpoints</a> schema
201 - Created	<a href="#">Endpoints</a> schema
401 - Unauthorized	Empty

### 10.2.6. /api/v1/watch/namespaces/{namespace}/endpoints/{name}

Table 10.19. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Endpoints

#### HTTP method

#### GET

#### Description

watch changes to an object of kind Endpoints. 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	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

## CHAPTER 11. ENDPOINTS LICE [DISCOVERY.K8S.IO/V1]

### Description

EndpointSlice represents a subset of the endpoints that implement a service. For a given service there may be multiple EndpointSlice objects, selected by labels, which must be joined to produce the full set of endpoints.

### Type

**object**

### Required

- **addressType**
- **endpoints**

## 11.1. SPECIFICATION

Property	Type	Description
<b>addressType</b>	<b>string</b>	<p>addressType specifies the type of address carried by this EndpointSlice. All addresses in this slice must be the same type. This field is immutable after creation. The following address types are currently supported:</p> <ul style="list-style-type: none"> <li>* IPv4: Represents an IPv4 Address.</li> <li>* IPv6: Represents an IPv6 Address.</li> <li>* FQDN: Represents a Fully Qualified Domain Name.</li> </ul> <p>Possible enum values: - <b>"FQDN"</b> represents a FQDN. - <b>"IPv4"</b> represents an IPv4 Address. - <b>"IPv6"</b> represents an IPv6 Address.</p>
<b>apiVersion</b>	<b>string</b>	<p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p>
<b>endpoints</b>	<b>array</b>	<p>endpoints is a list of unique endpoints in this slice. Each slice may include a maximum of 1000 endpoints.</p>

Property	Type	Description
<b>endpoints[]</b>	<b>object</b>	Endpoint represents a single logical "backend" implementing a service.
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata.
<b>ports</b>	<b>array</b>	ports specifies the list of network ports exposed by each endpoint in this slice. Each port must have a unique name. When ports is empty, it indicates that there are no defined ports. When a port is defined with a nil port value, it indicates "all ports". Each slice may include a maximum of 100 ports.
<b>ports[]</b>	<b>object</b>	EndpointPort represents a Port used by an EndpointSlice

### 11.1.1. .endpoints

#### Description

endpoints is a list of unique endpoints in this slice. Each slice may include a maximum of 1000 endpoints.

#### Type

**array**

### 11.1.2. .endpoints[]

#### Description

Endpoint represents a single logical "backend" implementing a service.

#### Type

**object**

#### Required

address

- **addresses**

Property	Type	Description
<b>addresses</b>	<b>array (string)</b>	addresses of this endpoint. The contents of this field are interpreted according to the corresponding EndpointSlice addressType field. Consumers must handle different types of addresses in the context of their own capabilities. This must contain at least one address but no more than 100. These are all assumed to be fungible and clients may choose to only use the first element. Refer to: <a href="https://issue.k8s.io/106267">https://issue.k8s.io/106267</a>
<b>conditions</b>	<b>object</b>	EndpointConditions represents the current condition of an endpoint.
<b>deprecatedTopology</b>	<b>object (string)</b>	deprecatedTopology contains topology information part of the v1beta1 API. This field is deprecated, and will be removed when the v1beta1 API is removed (no sooner than kubernetes v1.24). While this field can hold values, it is not writable through the v1 API, and any attempts to write to it will be silently ignored. Topology information can be found in the zone and nodeName fields instead.
<b>hints</b>	<b>object</b>	EndpointHints provides hints describing how an endpoint should be consumed.
<b>hostname</b>	<b>string</b>	hostname of this endpoint. This field may be used by consumers of endpoints to distinguish endpoints from each other (e.g. in DNS names). Multiple endpoints which use the same hostname should be considered fungible (e.g. multiple A values in DNS). Must be lowercase and pass DNS Label (RFC 1123) validation.

Property	Type	Description
<b>nodeName</b>	<b>string</b>	nodeName represents the name of the Node hosting this endpoint. This can be used to determine endpoints local to a Node.
<b>targetRef</b>	<b>ObjectReference</b>	targetRef is a reference to a Kubernetes object that represents this endpoint.
<b>zone</b>	<b>string</b>	zone is the name of the Zone this endpoint exists in.

### 11.1.3. .endpoints[].conditions

#### Description

EndpointConditions represents the current condition of an endpoint.

#### Type

**object**

Property	Type	Description
<b>ready</b>	<b>boolean</b>	ready indicates that this endpoint is prepared to receive traffic, according to whatever system is managing the endpoint. A nil value indicates an unknown state. In most cases consumers should interpret this unknown state as ready. For compatibility reasons, ready should never be "true" for terminating endpoints, except when the normal readiness behavior is being explicitly overridden, for example when the associated Service has set the publishNotReadyAddresses flag.
<b>serving</b>	<b>boolean</b>	serving is identical to ready except that it is set regardless of the terminating state of endpoints. This condition should be set to true for a ready endpoint that is terminating. If nil, consumers should defer to the ready condition.



Property	Type	Description
<b>terminating</b>	<b>boolean</b>	terminating indicates that this endpoint is terminating. A nil value indicates an unknown state. Consumers should interpret this unknown state to mean that the endpoint is not terminating.

#### 11.1.4. .endpoints[].hints

##### Description

EndpointHints provides hints describing how an endpoint should be consumed.

##### Type

**object**

Property	Type	Description
<b>forZones</b>	<b>array</b>	forZones indicates the zone(s) this endpoint should be consumed by to enable topology aware routing.
<b>forZones[]</b>	<b>object</b>	ForZone provides information about which zones should consume this endpoint.

#### 11.1.5. .endpoints[].hints.forZones

##### Description

forZones indicates the zone(s) this endpoint should be consumed by to enable topology aware routing.

##### Type

**array**

#### 11.1.6. .endpoints[].hints.forZones[]

##### Description

ForZone provides information about which zones should consume this endpoint.

##### Type

**object**

##### Required

- **name**

Property	Type	Description
<b>name</b>	<b>string</b>	name represents the name of the zone.

### 11.1.7. .ports

#### Description

ports specifies the list of network ports exposed by each endpoint in this slice. Each port must have a unique name. When ports is empty, it indicates that there are no defined ports. When a port is defined with a nil port value, it indicates "all ports". Each slice may include a maximum of 100 ports.

#### Type

**array**

### 11.1.8. .ports[]

#### Description

EndpointPort represents a Port used by an EndpointSlice

#### Type

**object**

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

Property	Type	Description
<b>appProtocol</b>	<b>string</b>	<p>The application protocol for this port. This is used as a hint for implementations to offer richer behavior for protocols that they understand. This field follows standard Kubernetes label syntax. Valid values are either:</p> <ul style="list-style-type: none"> <li>* Un-prefixed protocol names - reserved for IANA standard service names (as per RFC-6335 and <a href="https://www.iana.org/assignments/service-names">https://www.iana.org/assignments/service-names</a>).</li> <li>* Kubernetes-defined prefixed names: <ul style="list-style-type: none"> <li>* 'kubernetes.io/h2c' - HTTP/2 prior knowledge over cleartext as described in <a href="https://www.rfc-editor.org/rfc/rfc9113.html#name-starting-http-2-with-prior-">https://www.rfc-editor.org/rfc/rfc9113.html#name-starting-http-2-with-prior-</a></li> <li>* 'kubernetes.io/ws' - WebSocket over cleartext as described in <a href="https://www.rfc-editor.org/rfc/rfc6455">https://www.rfc-editor.org/rfc/rfc6455</a></li> <li>* 'kubernetes.io/wss' - WebSocket over TLS as described in <a href="https://www.rfc-editor.org/rfc/rfc6455">https://www.rfc-editor.org/rfc/rfc6455</a></li> </ul> </li> <li>* Other protocols should use implementation-defined prefixed names such as mycompany.com/my-custom-protocol.</li> </ul>

Property	Type	Description
<b>name</b>	<b>string</b>	name represents the name of this port. All ports in an EndpointSlice must have a unique name. If the EndpointSlice is derived from a Kubernetes service, this corresponds to the Service.ports[].name. Name must either be an empty string or pass DNS_LABEL validation: * must be no more than 63 characters long. * must consist of lower case alphanumeric characters or '-'. * must start and end with an alphanumeric character. Default is empty string.
<b>port</b>	<b>integer</b>	port represents the port number of the endpoint. If this is not specified, ports are not restricted and must be interpreted in the context of the specific consumer.
<b>protocol</b>	<b>string</b>	protocol represents the IP protocol for this port. Must be UDP, TCP, or SCTP. Default is TCP.  Possible enum values: - <b>"SCTP"</b> is the SCTP protocol. - <b>"TCP"</b> is the TCP protocol. - <b>"UDP"</b> is the UDP protocol.

## 11.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/discovery.k8s.io/v1/endpointslices**
  - **GET**: list or watch objects of kind EndpointSlice
- **/apis/discovery.k8s.io/v1/watch/endpointslices**
  - **GET**: watch individual changes to a list of EndpointSlice. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices**
  - **DELETE**: delete collection of EndpointSlice
  - **GET**: list or watch objects of kind EndpointSlice
  - **POST**: create an EndpointSlice

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

### 11.2.1. /apis/discovery.k8s.io/v1/endpointslices

HTTP method

**GET**

Description

list or watch objects of kind EndpointSlice

Table 11.1. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">EndpointSliceList</a> schema
401 - Unauthorized	Empty

### 11.2.2. /apis/discovery.k8s.io/v1/watch/endpointslices

HTTP method

**GET**

Description

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

Table 11.2. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

### 11.2.3. /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices

#### HTTP method

#### DELETE

#### Description

delete collection of EndpointSlice

Table 11.3. Query parameters

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

Table 11.4. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

list or watch objects of kind EndpointSlice

Table 11.5. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">EndpointSliceList</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### POST

#### Description

create an EndpointSlice

Table 11.6. Query parameters

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

Table 11.7. Body parameters

Parameter	Type	Description
<b>body</b>	<b>EndpointSlice</b> schema	

Table 11.8. HTTP responses

HTTP code	Response body
200 - OK	<b>EndpointSlice</b> schema
201 - Created	<b>EndpointSlice</b> schema
202 - Accepted	<b>EndpointSlice</b> schema
401 - Unauthorized	Empty

#### 11.2.4. /apis/discovery.k8s.io/v1/watch/namespaces/{namespace}/endpointslices

HTTP method

**GET****Description**

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

**Table 11.9. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

**11.2.5. /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices/{name}****Table 11.10. Global path parameters**

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EndpointSlice

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

delete an EndpointSlice

**Table 11.11. Query parameters**

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

**Table 11.12. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

**HTTP method**



**GET****Description**

read the specified EndpointSlice

**Table 11.13. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EndpointSlice</a> schema
401 - Unauthorized	Empty

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

partially update the specified EndpointSlice

**Table 11.14. Query parameters**

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

**Table 11.15. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EndpointSlice</a> schema
201 - Created	<a href="#">EndpointSlice</a> schema
401 - Unauthorized	Empty

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

replace the specified EndpointSlice

**Table 11.16. Query parameters**

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

**Table 11.17. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">EndpointSlice</a> schema	

Table 11.18. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">EndpointSlice</a> schema
201 - Created	<a href="#">EndpointSlice</a> schema
401 - Unauthorized	Empty

### 11.2.6. /apis/discovery.k8s.io/v1/watch/namespaces/{namespace}/endpointslices/{na

Table 11.19. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EndpointSlice

#### HTTP method

#### GET

#### Description

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

Table 11.20. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

## CHAPTER 12. EGRESSROUTER

### [NETWORK.OPERATOR.OPENSIFT.IO/V1]

#### Description

EgressRouter is a feature allowing the user to define an egress router that acts as a bridge between pods and external systems. The egress router runs a service that redirects egress traffic originating from a pod or a group of pods to a remote external system or multiple destinations as per configuration.

It is consumed by the cluster-network-operator. More specifically, given an EgressRouter CR with <name>, the CNO will create and manage:

- A service called <name>
- An egress pod called <name>
- A NAD called <name>

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

EgressRouter is a single egressrouter pod configuration object.

#### Type

**object**

#### Required

- **spec**

## 12.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>

Property	Type	Description
<b>metadata</b>	<a href="#">ObjectMeta</a>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	Specification of the desired egress router.
<b>status</b>	<b>object</b>	Observed status of EgressRouter.

### 12.1.1. .spec

#### Description

Specification of the desired egress router.

#### Type

**object**

#### Required

- **addresses**
- **mode**
- **networkInterface**

Property	Type	Description
<b>addresses</b>	<b>array</b>	List of IP addresses to configure on the pod's secondary interface.
<b>addresses[]</b>	<b>object</b>	EgressRouterAddress contains a pair of IP CIDR and gateway to be configured on the router's interface
<b>mode</b>	<b>string</b>	Mode depicts the mode that is used for the egress router. The default mode is "Redirect" and is the only supported mode currently.

Property	Type	Description
<b>networkInterface</b>	<b>object</b>	Specification of interface to create/use. The default is macvlan. Currently only macvlan is supported.
<b>redirect</b>	<b>object</b>	Redirect represents the configuration parameters specific to redirect mode.

### 12.1.2. .spec.addresses

#### Description

List of IP addresses to configure on the pod's secondary interface.

#### Type

**array**

### 12.1.3. .spec.addresses[]

#### Description

EgressRouterAddress contains a pair of IP CIDR and gateway to be configured on the router's interface

#### Type

**object**

#### Required

- **ip**

Property	Type	Description
<b>gateway</b>	<b>string</b>	IP address of the next-hop gateway, if it cannot be automatically determined. Can be IPv4 or IPv6.
<b>ip</b>	<b>string</b>	IP is the address to configure on the router's interface. Can be IPv4 or IPv6.

### 12.1.4. .spec.networkInterface

#### Description

Specification of interface to create/use. The default is macvlan. Currently only macvlan is supported.

#### Type

**object**

Property	Type	Description
<b>macvlan</b>	<b>object</b>	Arguments specific to the interfaceType macvlan

### 12.1.5. .spec.networkInterface.macvlan

#### Description

Arguments specific to the interfaceType macvlan

#### Type

**object**

#### Required

- **mode**

Property	Type	Description
<b>master</b>	<b>string</b>	Name of the master interface. Need not be specified if it can be inferred from the IP address.
<b>mode</b>	<b>string</b>	Mode depicts the mode that is used for the macvlan interface; one of Bridge Private VEPA Passthru. The default mode is "Bridge".

### 12.1.6. .spec.redirect

#### Description

Redirect represents the configuration parameters specific to redirect mode.

#### Type

**object**

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

Property	Type	Description
<b>fallbackIP</b>	<b>string</b>	FallbackIP specifies the remote destination's IP address. Can be IPv4 or IPv6. If no redirect rules are specified, all traffic from the router are redirected to this IP. If redirect rules are specified, then any connections on any other port (undefined in the rules) on the router will be redirected to this IP. If redirect rules are specified and no fallback IP is provided, connections on other ports will simply be rejected.
<b>redirectRules</b>	<b>array</b>	List of L4RedirectRules that define the DNAT redirection from the pod to the destination in redirect mode.
<b>redirectRules[]</b>	<b>object</b>	L4RedirectRule defines a DNAT redirection from a given port to a destination IP and port.

### 12.1.7. .spec.redirect.redirectRules

#### Description

List of L4RedirectRules that define the DNAT redirection from the pod to the destination in redirect mode.

#### Type

**array**

### 12.1.8. .spec.redirect.redirectRules[]

#### Description

L4RedirectRule defines a DNAT redirection from a given port to a destination IP and port.

#### Type

**object**

#### Required

- **destinationIP**
- **port**
- **protocol**



Property	Type	Description
<b>destinationIP</b>	<b>string</b>	IP specifies the remote destination's IP address. Can be IPv4 or IPv6.
<b>port</b>	<b>integer</b>	Port is the port number to which clients should send traffic to be redirected.
<b>protocol</b>	<b>string</b>	Protocol can be TCP, SCTP or UDP.
<b>targetPort</b>	<b>integer</b>	TargetPort allows specifying the port number on the remote destination to which the traffic gets redirected to. If unspecified, the value from "Port" is used.

### 12.1.9. .status

#### Description

Observed status of EgressRouter.

#### Type

**object**

#### Required

- **conditions**

Property	Type	Description
<b>conditions</b>	<b>array</b>	Observed status of the egress router
<b>conditions[]</b>	<b>object</b>	EgressRouterStatusCondition represents the state of the egress router's managed and monitored components.

### 12.1.10. .status.conditions

#### Description

Observed status of the egress router

#### Type

**array**

### 12.1.11. .status.conditions[]

## Description

EgressRouterStatusCondition represents the state of the egress router's managed and monitored components.

## Type

**object**

## Required

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

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

## 12.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/network.operator.openshift.io/v1/egressrouters**
  - **GET**: list objects of kind EgressRouter
- **/apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrouters**
  - **DELETE**: delete collection of EgressRouter
  - **GET**: list objects of kind EgressRouter
  - **POST**: create an EgressRouter

- **/apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrouters/{name}**
  - **DELETE**: delete an EgressRouter
  - **GET**: read the specified EgressRouter
  - **PATCH**: partially update the specified EgressRouter
  - **PUT**: replace the specified EgressRouter
- **/apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrouters/{name}/status**
  - **GET**: read status of the specified EgressRouter
  - **PATCH**: partially update status of the specified EgressRouter
  - **PUT**: replace status of the specified EgressRouter

### 12.2.1. /apis/network.operator.openshift.io/v1/egressrouters

HTTP method

**GET**

Description

list objects of kind EgressRouter

Table 12.1. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">EgressRouterList</a> schema
401 - Unauthorized	Empty

### 12.2.2. /apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrouters

HTTP method

**DELETE**

Description

delete collection of EgressRouter

Table 12.2. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

HTTP method

**GET****Description**

list objects of kind EgressRouter

**Table 12.3. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">EgressRouterList</a> schema
401 - Unauthorized	Empty

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

create an EgressRouter

**Table 12.4. Query parameters**

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

**Table 12.5. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>EgressRouter</b> schema	

Table 12.6. HTTP responses

HTTP code	Reponse body
200 - OK	<b>EgressRouter</b> schema
201 - Created	<b>EgressRouter</b> schema
202 - Accepted	<b>EgressRouter</b> schema
401 - Unauthorized	Empty

### 12.2.3. /apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrouter

Table 12.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressRouter

#### HTTP method

#### DELETE

#### Description

delete an EgressRouter

Table 12.8. Query parameters

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

Table 12.9. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema

HTTP code	Reponse body
401 - Unauthorized	Empty

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

read the specified EgressRouter

**Table 12.10. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">EgressRouter</a> schema
401 - Unauthorized	Empty

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

partially update the specified EgressRouter

**Table 12.11. Query parameters**

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

Table 12.12. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">EgressRouter</a> schema
401 - Unauthorized	Empty

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

replace the specified EgressRouter

Table 12.13. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 12.14. Body parameters

Parameter	Type	Description
<b>body</b>	<b>EgressRouter</b> schema	

Table 12.15. HTTP responses

HTTP code	Response body
200 - OK	<b>EgressRouter</b> schema
201 - Created	<b>EgressRouter</b> schema
401 - Unauthorized	Empty

#### 12.2.4. /apis/network.operator.openshift.io/v1/namespaces/{namespace}/egressrout

Table 12.16. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the EgressRouter

HTTP method

**GET**



**Description**

read status of the specified EgressRouter

**Table 12.17. HTTP responses**

HTTP code	Response body
200 - OK	<b>EgressRouter</b> schema
401 - Unauthorized	Empty

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

partially update status of the specified EgressRouter

**Table 12.18. Query parameters**

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

**Table 12.19. HTTP responses**

HTTP code	Response body
200 - OK	<b>EgressRouter</b> schema
401 - Unauthorized	Empty

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

replace status of the specified EgressRouter

**Table 12.20. Query parameters**

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

**Table 12.21. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>EgressRouter</b> schema	

**Table 12.22. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>EgressRouter</b> schema
201 - Created	<b>EgressRouter</b> schema
401 - Unauthorized	Empty

## CHAPTER 13. INGRESS [NETWORKING.K8S.IO/V1]

### Description

Ingress is a collection of rules that allow inbound connections to reach the endpoints defined by a backend. An Ingress can be configured to give services externally-reachable urls, load balance traffic, terminate SSL, offer name based virtual hosting etc.

### Type

**object**

### 13.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	IngressSpec describes the Ingress the user wishes to exist.
<b>status</b>	<b>object</b>	IngressStatus describe the current state of the Ingress.

#### 13.1.1. .spec

**Description**

IngressSpec describes the Ingress the user wishes to exist.

**Type**

**object**

Property	Type	Description
<b>defaultBackend</b>	<b>object</b>	IngressBackend describes all endpoints for a given service and port.
<b>ingressClassName</b>	<b>string</b>	ingressClassName is the name of an IngressClass cluster resource. Ingress controller implementations use this field to know whether they should be serving this Ingress resource, by a transitive connection (controller → IngressClass → Ingress resource). Although the <b>kubernetes.io/ingress.class</b> annotation (simple constant name) was never formally defined, it was widely supported by Ingress controllers to create a direct binding between Ingress controller and Ingress resources. Newly created Ingress resources should prefer using the field. However, even though the annotation is officially deprecated, for backwards compatibility reasons, ingress controllers should still honor that annotation if present.
<b>rules</b>	<b>array</b>	rules is a list of host rules used to configure the Ingress. If unspecified, or no rule matches, all traffic is sent to the default backend.
<b>rules[]</b>	<b>object</b>	IngressRule represents the rules mapping the paths under a specified host to the related backend services. Incoming requests are first evaluated for a host match, then routed to the backend associated with the matching IngressRuleValue.

Property	Type	Description
<b>tls</b>	<b>array</b>	tls represents the TLS configuration. Currently the Ingress only supports a single TLS port, 443. If multiple members of this list specify different hosts, they will be multiplexed on the same port according to the hostname specified through the SNI TLS extension, if the ingress controller fulfilling the ingress supports SNI.
<b>tls[]</b>	<b>object</b>	IngressTLS describes the transport layer security associated with an ingress.

### 13.1.2. .spec.defaultBackend

#### Description

IngressBackend describes all endpoints for a given service and port.

#### Type

**object**

Property	Type	Description
<b>resource</b>	<b>TypedLocalObjectReference</b>	resource is an ObjectRef to another Kubernetes resource in the namespace of the Ingress object. If resource is specified, a service.Name and service.Port must not be specified. This is a mutually exclusive setting with "Service".
<b>service</b>	<b>object</b>	IngressServiceBackend references a Kubernetes Service as a Backend.

### 13.1.3. .spec.defaultBackend.service

#### Description

IngressServiceBackend references a Kubernetes Service as a Backend.

#### Type

**object**

#### Required

- **name**

Property	Type	Description
<b>name</b>	<b>string</b>	name is the referenced service. The service must exist in the same namespace as the Ingress object.
<b>port</b>	<b>object</b>	ServiceBackendPort is the service port being referenced.

#### 13.1.4. .spec.defaultBackend.service.port

##### Description

ServiceBackendPort is the service port being referenced.

##### Type

**object**

Property	Type	Description
<b>name</b>	<b>string</b>	name is the name of the port on the Service. This is a mutually exclusive setting with "Number".
<b>number</b>	<b>integer</b>	number is the numerical port number (e.g. 80) on the Service. This is a mutually exclusive setting with "Name".

#### 13.1.5. .spec.rules

##### Description

rules is a list of host rules used to configure the Ingress. If unspecified, or no rule matches, all traffic is sent to the default backend.

##### Type

**array**

#### 13.1.6. .spec.rules[]

##### Description

IngressRule represents the rules mapping the paths under a specified host to the related backend services. Incoming requests are first evaluated for a host match, then routed to the backend associated with the matching IngressRuleValue.

##### Type

**object**

Property	Type	Description
<b>host</b>	<b>string</b>	<p>host is the fully qualified domain name of a network host, as defined by RFC 3986. Note the following deviations from the "host" part of the URI as defined in RFC 3986: 1. IPs are not allowed. Currently an IngressRuleValue can only apply to the IP in the Spec of the parent Ingress. 2. The <code>:</code> delimiter is not respected because ports are not allowed. Currently the port of an Ingress is implicitly <code>:80</code> for http and <code>:443</code> for https. Both these may change in the future. Incoming requests are matched against the host before the IngressRuleValue. If the host is unspecified, the Ingress routes all traffic based on the specified IngressRuleValue.</p> <p>host can be "precise" which is a domain name without the terminating dot of a network host (e.g. "foo.bar.com") or "wildcard", which is a domain name prefixed with a single wildcard label (e.g. ".foo.com"). <b>The wildcard character "</b> must appear by itself as the first DNS label and matches only a single label. You cannot have a wildcard label by itself (e.g. Host == "*"). Requests will be matched against the Host field in the following way: 1. If host is precise, the request matches this rule if the http host header is equal to Host. 2. If host is a wildcard, then the request matches this rule if the http host header is to equal to the suffix (removing the first label) of the wildcard rule.</p>



Property	Type	Description
<b>http</b>	<b>object</b>	HTTPIngressRuleValue is a list of http selectors pointing to backends. In the example: <a href="http://&lt;host&gt;/&lt;path&gt;?&lt;searchpart&gt;">http://&lt;host&gt;/&lt;path&gt;?&lt;searchpart&gt;</a> → backend where where parts of the url correspond to RFC 3986, this resource will be used to match against everything after the last '/' and before the first '?' or '#'.

### 13.1.7. .spec.rules[].http

#### Description

HTTPIngressRuleValue is a list of http selectors pointing to backends. In the example: <http://<host>/<path>?<searchpart>> → backend where where parts of the url correspond to RFC 3986, this resource will be used to match against everything after the last '/' and before the first '?' or '#'.

#### Type

**object**

#### Required

- **paths**

Property	Type	Description
<b>paths</b>	<b>array</b>	paths is a collection of paths that map requests to backends.
<b>paths[]</b>	<b>object</b>	HTTPIngressPath associates a path with a backend. Incoming urls matching the path are forwarded to the backend.

### 13.1.8. .spec.rules[].http.paths

#### Description

paths is a collection of paths that map requests to backends.

#### Type

**array**

### 13.1.9. .spec.rules[].http.paths[]

#### Description

HTTPIngressPath associates a path with a backend. Incoming urls matching the path are forwarded to the backend.

## Type

**object**

## Required

- **pathType**
- **backend**

Property	Type	Description
<b>backend</b>	<b>object</b>	IngressBackend describes all endpoints for a given service and port.
<b>path</b>	<b>string</b>	path is matched against the path of an incoming request. Currently it can contain characters disallowed from the conventional "path" part of a URL as defined by RFC 3986. Paths must begin with a '/' and must be present when using PathType with value "Exact" or "Prefix".
<b>pathType</b>	<b>string</b>	<p>pathType determines the interpretation of the path matching. PathType can be one of the following values: *</p> <ul style="list-style-type: none"> <li>Exact: Matches the URL path exactly.</li> <li>Prefix: Matches based on a URL path prefix split by '/'. Matching is done on a path element by element basis. A path element refers is the list of labels in the path split by the '/' separator. A request is a match for path p if every p is an element-wise prefix of p of the request path. Note that if the last element of the path is a substring of the last element in request path, it is not a match (e.g. /foo/bar matches /foo/bar/baz, but does not match /foo/barbaz).</li> </ul> <p>* ImplementationSpecific: Interpretation of the Path matching is up to the IngressClass. Implementations can treat this as a separate PathType or treat it identically to Prefix or Exact path types. Implementations are required to support all path types.</p>

Property	Type	Possible enum values: - <b>"Exact"</b> <b>Description</b> matches the URL path exactly and with case sensitivity. - <b>"ImplementationSpecific"</b> matching is up to the IngressClass. Implementations can treat this as a separate PathType or treat it identically to Prefix or Exact path types. - <b>"Prefix"</b> matches based on a URL path prefix split by '/'. Matching is case sensitive and done on a path element by element basis. A path element refers to the list of labels in the path split by the '/' separator. A request is a match for path p if every p is an element-wise prefix of p of the request path. Note that if the last element of the path is a substring of the last element in request path, it is not a match (e.g. /foo/bar matches /foo/bar/baz, but does not match /foo/barbaz). If multiple matching paths exist in an Ingress spec, the longest matching path is given priority. Examples: - /foo/bar does not match requests to /foo/barbaz - /foo/bar matches request to /foo/bar and /foo/bar/baz - /foo and /foo/ both match requests to /foo and /foo/. If both paths are present in an Ingress spec, the longest matching path (/foo/) is given priority.

### 13.1.10. .spec.rules[].http.paths[].backend

#### Description

IngressBackend describes all endpoints for a given service and port.

#### Type

**object**

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

Property	Type	Description
<b>resource</b>	<b>TypedLocalObjectReference</b>	resource is an ObjectRef to another Kubernetes resource in the namespace of the Ingress object. If resource is specified, a service.Name and service.Port must not be specified. This is a mutually exclusive setting with "Service".
<b>service</b>	<b>object</b>	IngressServiceBackend references a Kubernetes Service as a Backend.

### 13.1.11. .spec.rules[].http.paths[].backend.service

#### Description

IngressServiceBackend references a Kubernetes Service as a Backend.

#### Type

**object**

#### Required

- **name**

Property	Type	Description
<b>name</b>	<b>string</b>	name is the referenced service. The service must exist in the same namespace as the Ingress object.
<b>port</b>	<b>object</b>	ServiceBackendPort is the service port being referenced.

### 13.1.12. .spec.rules[].http.paths[].backend.service.port

#### Description

ServiceBackendPort is the service port being referenced.

#### Type

**object**

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

Property	Type	Description
<b>name</b>	<b>string</b>	name is the name of the port on the Service. This is a mutually exclusive setting with "Number".
<b>number</b>	<b>integer</b>	number is the numerical port number (e.g. 80) on the Service. This is a mutually exclusive setting with "Name".

### 13.1.13. .spec.tls

#### Description

tls represents the TLS configuration. Currently the Ingress only supports a single TLS port, 443. If multiple members of this list specify different hosts, they will be multiplexed on the same port according to the hostname specified through the SNI TLS extension, if the ingress controller fulfilling the ingress supports SNI.

#### Type

**array**

### 13.1.14. .spec.tls[]

#### Description

IngressTLS describes the transport layer security associated with an ingress.

#### Type

**object**

Property	Type	Description
<b>hosts</b>	<b>array (string)</b>	hosts is a list of hosts included in the TLS certificate. The values in this list must match the name/s used in the tlsSecret. Defaults to the wildcard host setting for the loadbalancer controller fulfilling this Ingress, if left unspecified.
<b>secretName</b>	<b>string</b>	secretName is the name of the secret used to terminate TLS traffic on port 443. Field is left optional to allow TLS routing based on SNI hostname alone. If the SNI host in a listener conflicts with the "Host" header field used by an IngressRule, the SNI host is used for termination and value of the "Host" header is used for routing.

### 13.1.15. .status

#### Description

IngressStatus describe the current state of the Ingress.

#### Type

**object**

Property	Type	Description
<b>loadBalancer</b>	<b>object</b>	IngressLoadBalancerStatus represents the status of a load-balancer.

### 13.1.16. .status.loadBalancer

#### Description

IngressLoadBalancerStatus represents the status of a load-balancer.

#### Type

**object**

Property	Type	Description
<b>ingress</b>	<b>array</b>	ingress is a list containing ingress points for the load-balancer.
<b>ingress[]</b>	<b>object</b>	IngressLoadBalancerIngress represents the status of a load-balancer ingress point.

### 13.1.17. .status.loadBalancer.ingress

#### Description

ingress is a list containing ingress points for the load-balancer.

#### Type

**array**

### 13.1.18. .status.loadBalancer.ingress[]

#### Description

IngressLoadBalancerIngress represents the status of a load-balancer ingress point.

#### Type

**object**

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

Property	Type	Description
<b>hostname</b>	<b>string</b>	hostname is set for load-balancer ingress points that are DNS based.
<b>ip</b>	<b>string</b>	ip is set for load-balancer ingress points that are IP based.
<b>ports</b>	<b>array</b>	ports provides information about the ports exposed by this LoadBalancer.
<b>ports[]</b>	<b>object</b>	IngressPortStatus represents the error condition of a service port

### 13.1.19. .status.loadBalancer.ingress[].ports

#### Description

ports provides information about the ports exposed by this LoadBalancer.

#### Type

**array**

### 13.1.20. .status.loadBalancer.ingress[].ports[]

#### Description

IngressPortStatus represents the error condition of a service port

#### Type

**object**

#### Required

- **port**
- **protocol**

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

Property	Type	Description
<b>error</b>	<b>string</b>	error is to record the problem with the service port. The format of the error shall comply with the following rules: - built-in error values shall be specified in this file and those shall use CamelCase names - cloud provider specific error values must have names that comply with the format foo.example.com/CamelCase.
<b>port</b>	<b>integer</b>	port is the port number of the ingress port.
<b>protocol</b>	<b>string</b>	protocol is the protocol of the ingress port. The supported values are: "TCP", "UDP", "SCTP"  Possible enum values: - <b>"SCTP"</b> is the SCTP protocol. - <b>"TCP"</b> is the TCP protocol. - <b>"UDP"</b> is the UDP protocol.

## 13.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/networking.k8s.io/v1/ingresses**
  - **GET**: list or watch objects of kind Ingress
- **/apis/networking.k8s.io/v1/watch/ingresses**
  - **GET**: watch individual changes to a list of Ingress. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses**
  - **DELETE**: delete collection of Ingress
  - **GET**: list or watch objects of kind Ingress
  - **POST**: create an Ingress
- **/apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses**
  - **GET**: watch individual changes to a list of Ingress. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}**
  - **DELETE**: delete an Ingress



- **GET**: read the specified Ingress
- **PATCH**: partially update the specified Ingress
- **PUT**: replace the specified Ingress
- **/apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses/{name}**
  - **GET**: watch changes to an object of kind Ingress. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status**
  - **GET**: read status of the specified Ingress
  - **PATCH**: partially update status of the specified Ingress
  - **PUT**: replace status of the specified Ingress

### 13.2.1. /apis/networking.k8s.io/v1/ingresses

HTTP method

**GET**

Description

list or watch objects of kind Ingress

Table 13.1. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">IngressList</a> schema
401 - Unauthorized	Empty

### 13.2.2. /apis/networking.k8s.io/v1/watch/ingresses

HTTP method

**GET**

Description

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

Table 13.2. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

### 13.2.3. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses

#### HTTP method

#### DELETE

#### Description

delete collection of Ingress

Table 13.3. Query parameters

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

Table 13.4. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

list or watch objects of kind Ingress

Table 13.5. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">IngressList</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### POST

#### Description

create an Ingress

Table 13.6. Query parameters

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

Table 13.7. Body parameters

Parameter	Type	Description
<b>body</b>	<b>Ingress</b> schema	

Table 13.8. HTTP responses

HTTP code	Response body
200 - OK	<b>Ingress</b> schema
201 - Created	<b>Ingress</b> schema
202 - Accepted	<b>Ingress</b> schema
401 - Unauthorized	Empty

#### 13.2.4. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses

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

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

**Table 13.9. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

**13.2.5. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}****Table 13.10. Global path parameters**

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Ingress

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

delete an Ingress

**Table 13.11. Query parameters**

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

**Table 13.12. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

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

read the specified Ingress

**Table 13.13. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">Ingress</a> schema
401 - Unauthorized	Empty

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

partially update the specified Ingress

**Table 13.14. Query parameters**

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

**Table 13.15. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">Ingress</a> schema
201 - Created	<a href="#">Ingress</a> schema
401 - Unauthorized	Empty

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

replace the specified Ingress

**Table 13.16. Query parameters**

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

**Table 13.17. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">Ingress</a> schema	

Table 13.18. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Ingress</a> schema
201 - Created	<a href="#">Ingress</a> schema
401 - Unauthorized	Empty

### 13.2.6. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses/{name}

Table 13.19. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Ingress

#### HTTP method

#### GET

#### Description

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

Table 13.20. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

### 13.2.7. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status

Table 13.21. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Ingress

#### HTTP method

#### GET

#### Description

read status of the specified Ingress

Table 13.22. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">Ingress</a> schema
401 - Unauthorized	Empty

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

partially update status of the specified Ingress

**Table 13.23. Query parameters**

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

**Table 13.24. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">Ingress</a> schema
201 - Created	<a href="#">Ingress</a> schema



HTTP code	Response body
401 - Unauthorized	Empty

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

replace status of the specified Ingress

**Table 13.25. Query parameters**

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

**Table 13.26. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>Ingress</b> schema	

**Table 13.27. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>Ingress</b> schema
201 - Created	<b>Ingress</b> schema
401 - Unauthorized	Empty

## CHAPTER 14. INGRESSCLASS [NETWORKING.K8S.IO/V1]

### Description

IngressClass represents the class of the Ingress, referenced by the Ingress Spec. The **ingressclass.kubernetes.io/is-default-class** annotation can be used to indicate that an IngressClass should be considered default. When a single IngressClass resource has this annotation set to true, new Ingress resources without a class specified will be assigned this default class.

### Type

**object**

## 14.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	IngressClassSpec provides information about the class of an Ingress.

### 14.1.1. .spec

#### Description

IngressClassSpec provides information about the class of an Ingress.

## Type

**object**

Property	Type	Description
<b>controller</b>	<b>string</b>	controller refers to the name of the controller that should handle this class. This allows for different "flavors" that are controlled by the same controller. For example, you may have different parameters for the same implementing controller. This should be specified as a domain-prefixed path no more than 250 characters in length, e.g. "acme.io/ingress-controller". This field is immutable.
<b>parameters</b>	<b>object</b>	IngressClassParametersReference identifies an API object. This can be used to specify a cluster or namespace-scoped resource.

### 14.1.2. .spec.parameters

#### Description

IngressClassParametersReference identifies an API object. This can be used to specify a cluster or namespace-scoped resource.

## Type

**object**

#### Required

- **kind**
- **name**

Property	Type	Description
<b>apiGroup</b>	<b>string</b>	apiGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<b>kind</b>	<b>string</b>	kind is the type of resource being referenced.

Property	Type	Description
<b>name</b>	<b>string</b>	name is the name of resource being referenced.
<b>namespace</b>	<b>string</b>	namespace is the namespace of the resource being referenced. This field is required when scope is set to "Namespace" and must be unset when scope is set to "Cluster".
<b>scope</b>	<b>string</b>	scope represents if this refers to a cluster or namespace scoped resource. This may be set to "Cluster" (default) or "Namespace".

## 14.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/networking.k8s.io/v1/ingressclasses**
  - **DELETE**: delete collection of IngressClass
  - **GET**: list or watch objects of kind IngressClass
  - **POST**: create an IngressClass
- **/apis/networking.k8s.io/v1/watch/ingressclasses**
  - **GET**: watch individual changes to a list of IngressClass. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/networking.k8s.io/v1/ingressclasses/{name}**
  - **DELETE**: delete an IngressClass
  - **GET**: read the specified IngressClass
  - **PATCH**: partially update the specified IngressClass
  - **PUT**: replace the specified IngressClass
- **/apis/networking.k8s.io/v1/watch/ingressclasses/{name}**
  - **GET**: watch changes to an object of kind IngressClass. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

### 14.2.1. /apis/networking.k8s.io/v1/ingressclasses

HTTP method

**DELETE****Description**

delete collection of IngressClass

**Table 14.1. Query parameters**

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

**Table 14.2. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
401 - Unauthorized	Empty

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

list or watch objects of kind IngressClass

**Table 14.3. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>IngressClassList</b> schema
401 - Unauthorized	Empty

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

create an IngressClass

**Table 14.4. Query parameters**

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

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

Table 14.5. Body parameters

Parameter	Type	Description
<b>body</b>	<b>IngressClass</b> schema	

Table 14.6. HTTP responses

HTTP code	Response body
200 - OK	<b>IngressClass</b> schema
201 - Created	<b>IngressClass</b> schema
202 - Accepted	<b>IngressClass</b> schema
401 - Unauthorized	Empty

### 14.2.2. /apis/networking.k8s.io/v1/watch/ingressclasses

#### HTTP method

**GET****Description**

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

**Table 14.7. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

**14.2.3. /apis/networking.k8s.io/v1/ingressclasses/{name}****Table 14.8. Global path parameters**

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the IngressClass

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

delete an IngressClass

**Table 14.9. Query parameters**

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

**Table 14.10. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

**HTTP method**



**GET****Description**

read the specified IngressClass

**Table 14.11. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">IngressClass</a> schema
401 - Unauthorized	Empty

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

partially update the specified IngressClass

**Table 14.12. Query parameters**

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

**Table 14.13. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">IngressClass</a> schema
201 - Created	<a href="#">IngressClass</a> schema
401 - Unauthorized	Empty

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

replace the specified IngressClass

**Table 14.14. Query parameters**

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

**Table 14.15. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">IngressClass</a> schema	

Table 14.16. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">IngressClass</a> schema
201 - Created	<a href="#">IngressClass</a> schema
401 - Unauthorized	Empty

#### 14.2.4. /apis/networking.k8s.io/v1/watch/ingressclasses/{name}

Table 14.17. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the IngressClass

#### HTTP method

##### GET

#### Description

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

Table 14.18. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

## CHAPTER 15. IPPOOL

### [WHEREABOUTS.CNI.CNCF.IO/V1ALPHA1]

#### Description

IPPool is the Schema for the ippools API

#### Type

**object**

## 15.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	IPPoolSpec defines the desired state of IPPool

### 15.1.1. .spec

#### Description

IPPoolSpec defines the desired state of IPPool

#### Type

**object****Required**

- **allocations**
- **range**

Property	Type	Description
<b>allocations</b>	<b>object</b>	Allocations is the set of allocated IPs for the given range. Its indices are a direct mapping to the IP with the same index/offset for the pool's range.
<b>allocations{}</b>	<b>object</b>	IPAllocation represents metadata about the pod/container owner of a specific IP
<b>range</b>	<b>string</b>	Range is a RFC 4632/4291-style string that represents an IP address and prefix length in CIDR notation

**15.1.2. .spec.allocations****Description**

Allocations is the set of allocated IPs for the given range. Its indices are a direct mapping to the IP with the same index/offset for the pool's range.

**Type****object****15.1.3. .spec.allocations{}****Description**

IPAllocation represents metadata about the pod/container owner of a specific IP

**Type****object****Required**

- **id**
- **podref**

Property	Type	Description
<b>id</b>	<b>string</b>	

Property	Type	Description
<b>ifname</b>	<b>string</b>	
<b>podref</b>	<b>string</b>	

## 15.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/whereabouts.cni.cncf.io/v1alpha1/ippools**
  - **GET**: list objects of kind IPPool
- **/apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/ippools**
  - **DELETE**: delete collection of IPPool
  - **GET**: list objects of kind IPPool
  - **POST**: create an IPPool
- **/apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/ippools/{name}**
  - **DELETE**: delete an IPPool
  - **GET**: read the specified IPPool
  - **PATCH**: partially update the specified IPPool
  - **PUT**: replace the specified IPPool

### 15.2.1. /apis/whereabouts.cni.cncf.io/v1alpha1/ippools

HTTP method

**GET**

Description

list objects of kind IPPool

Table 15.1. HTTP responses

HTTP code	Reponse body
200 - OK	<b>IPPoolList</b> schema
401 - Unauthorized	Empty

### 15.2.2. /apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/ippools

HTTP method

**DELETE****Description**

delete collection of IPPool

**Table 15.2. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

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

list objects of kind IPPool

**Table 15.3. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">IPPoolList</a> schema
401 - Unauthorized	Empty

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

create an IPPool

**Table 15.4. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 15.5. Body parameters

Parameter	Type	Description
<b>body</b>	<b>IPPool</b> schema	

Table 15.6. HTTP responses

HTTP code	Response body
200 - OK	<b>IPPool</b> schema
201 - Created	<b>IPPool</b> schema
202 - Accepted	<b>IPPool</b> schema
401 - Unauthorized	Empty

### 15.2.3. /apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/ippools/{na

Table 15.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the IPPool



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

delete an IPPool

**Table 15.8. Query parameters**

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

**Table 15.9. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

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

read the specified IPPool

**Table 15.10. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>IPPool</b> schema
401 - Unauthorized	Empty

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

partially update the specified IPPool

**Table 15.11. Query parameters**

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

Table 15.12. HTTP responses

HTTP code	Response body
200 - OK	<b>IPPool</b> schema
401 - Unauthorized	Empty

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

replace the specified IPPool

Table 15.13. Query parameters

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

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

Table 15.14. Body parameters

Parameter	Type	Description
<b>body</b>	<b>IPPool</b> schema	

Table 15.15. HTTP responses

HTTP code	Reponse body
200 - OK	<b>IPPool</b> schema
201 - Created	<b>IPPool</b> schema
401 - Unauthorized	Empty

## CHAPTER 16. MULTINETWORKPOLICY

### [K8S.CNI.CNCF.IO/V1BETA1]

#### Description

MultiNetworkPolicy is a CRD schema to provide NetworkPolicy mechanism for net-attach-def which is specified by the Network Plumbing Working Group. MultiNetworkPolicy is identical to Kubernetes NetworkPolicy, See: <https://kubernetes.io/docs/concepts/services-networking/network-policies/>.

#### Type

**object**

### 16.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	Specification of the desired behavior for this MultiNetworkPolicy.

#### 16.1.1. .spec

**Description**

Specification of the desired behavior for this MultiNetworkPolicy.

**Type**

**object**

**Required**

- **podSelector**

Property	Type	Description
<b>egress</b>	<b>array</b>	List of egress rules to be applied to the selected pods. Outgoing traffic is allowed if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic matches at least one egress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy limits all outgoing traffic (and serves solely to ensure that the pods it selects are isolated by default). This field is beta-level in 1.8
<b>egress[]</b>	<b>object</b>	NetworkPolicyEgressRule describes a particular set of traffic that is allowed out of pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and to. This type is beta-level in 1.8
<b>ingress</b>	<b>array</b>	List of ingress rules to be applied to the selected pods. Traffic is allowed to a pod if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic source is the pod's local node, OR if the traffic matches at least one ingress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy does not allow any traffic (and serves solely to ensure that the pods it selects are isolated by default)

Property	Type	Description
<b>ingress[]</b>	<b>object</b>	NetworkPolicyIngressRule describes a particular set of traffic that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and from.
<b>podSelector</b>	<b>object</b>	This is a label selector which selects Pods. This field follows standard label selector semantics; if present but empty, it selects all pods. If NamespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the Pods matching PodSelector in the policy's own Namespace.
<b>policyTypes</b>	<b>array (string)</b>	List of rule types that the NetworkPolicy relates to. Valid options are 'Ingress', 'Egress', or 'Ingress,Egress'. If this field is not specified, it will default based on the existence of Ingress or Egress rules; policies that contain an Egress section are assumed to affect Egress, and all policies (whether or not they contain an Ingress section) are assumed to affect Ingress. If you want to write an egress-only policy, you must explicitly specify policyTypes [ 'Egress' ]. Likewise, if you want to write a policy that specifies that no egress is allowed, you must specify a policyTypes value that include 'Egress' (since such a policy would not include an Egress section and would otherwise default to just [ 'Ingress' ]). This field is beta-level in 1.8

### 16.1.2. .spec.egress

#### Description

List of egress rules to be applied to the selected pods. Outgoing traffic is allowed if there are no

NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic matches at least one egress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy limits all outgoing traffic (and serves solely to ensure that the pods it selects are isolated by default). This field is beta-level in 1.8

#### Type

**array**

### 16.1.3. .spec.egress[]

#### Description

NetworkPolicyEgressRule describes a particular set of traffic that is allowed out of pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and to. This type is beta-level in 1.8

#### Type

**object**

Property	Type	Description
<b>ports</b>	<b>array</b>	List of destination ports for outgoing traffic. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.
<b>ports[]</b>	<b>object</b>	NetworkPolicyPort describes a port to allow traffic on
<b>to</b>	<b>array</b>	List of destinations for outgoing traffic of pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all destinations (traffic not restricted by destination). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the to list.
<b>to[]</b>	<b>object</b>	NetworkPolicyPeer describes a peer to allow traffic from. Only certain combinations of fields are allowed

### 16.1.4. .spec.egress[].ports

#### Description

List of destination ports for outgoing traffic. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.

#### Type

**array**

### 16.1.5. .spec.egress[].ports[]

#### Description

NetworkPolicyPort describes a port to allow traffic on

#### Type

**object**

Property	Type	Description
<b>port</b>	<b>integer-or-string</b>	The port on the given protocol. This can either be a numerical or named port on a pod. If this field is not provided, this matches all port names and numbers.
<b>protocol</b>	<b>string</b>	The protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.

### 16.1.6. .spec.egress[].to

#### Description

List of destinations for outgoing traffic of pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all destinations (traffic not restricted by destination). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the to list.

#### Type

**array**

### 16.1.7. .spec.egress[].to[]

#### Description

NetworkPolicyPeer describes a peer to allow traffic from. Only certain combinations of fields are allowed

#### Type

**object**



Property	Type	Description
<b>ipBlock</b>	<b>object</b>	IPBlock defines policy on a particular IPBlock. If this field is set then neither of the other fields can be.
<b>namespaceSelector</b>	<b>object</b>	Selects Namespaces using cluster-scoped labels. This field follows standard label selector semantics; if present but empty, it selects all namespaces. If PodSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects all Pods in the Namespaces selected by NamespaceSelector.
<b>podSelector</b>	<b>object</b>	This is a label selector which selects Pods. This field follows standard label selector semantics; if present but empty, it selects all pods. If NamespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the Pods matching PodSelector in the policy's own Namespace.

### 16.1.8. .spec.egress[.to[.ipBlock

#### Description

IPBlock defines policy on a particular IPBlock. If this field is set then neither of the other fields can be.

#### Type

**object**

#### Required

- **cidr**

Property	Type	Description
<b>cidr</b>	<b>string</b>	CIDR is a string representing the IP Block Valid examples are '192.168.1.1/24'

Property	Type	Description
<b>except</b>	<b>array (string)</b>	Except is a slice of CIDRs that should not be included within an IP Block Valid examples are '192.168.1.1/24' Except values will be rejected if they are outside the CIDR range

### 16.1.9. .spec.egress[].to[].namespaceSelector

#### Description

Selects Namespaces using cluster-scoped labels. This field follows standard label selector semantics; if present but empty, it selects all namespaces. If PodSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects all Pods in the Namespaces selected by NamespaceSelector.

#### Type

**object**

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

### 16.1.10. .spec.egress[].to[].namespaceSelector.matchExpressions

#### Description

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

#### Type

**array**

### 16.1.11. .spec.egress[].to[].namespaceSelector.matchExpressions[]

**Description**

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

**Type**

**object**

**Required**

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

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

**16.1.12. .spec.egress[].to[].podSelector****Description**

This is a label selector which selects Pods. This field follows standard label selector semantics; if present but empty, it selects all pods. If NamespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the Pods matching PodSelector in the policy's own Namespace.

**Type**

**object**

Property	Type	Description
<b>matchExpressions</b>	<b>array</b>	matchExpressions is a list of label selector requirements. The requirements are ANDed.

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

### 16.1.13. .spec.egress[].to[].podSelector.matchExpressions

#### Description

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

#### Type

**array**

### 16.1.14. .spec.egress[].to[].podSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

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

### 16.1.15. .spec.ingress

#### Description

List of ingress rules to be applied to the selected pods. Traffic is allowed to a pod if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic source is the pod's local node, OR if the traffic matches at least one ingress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy does not allow any traffic (and serves solely to ensure that the pods it selects are isolated by default)

#### Type

**array**

### 16.1.16. .spec.ingress[]

#### Description

NetworkPolicyIngressRule describes a particular set of traffic that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and from.

#### Type

**object**

Property	Type	Description
<b>from</b>	<b>array</b>	List of sources which should be able to access the pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all sources (traffic not restricted by source). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the from list.
<b>from[]</b>	<b>object</b>	NetworkPolicyPeer describes a peer to allow traffic from. Only certain combinations of fields are allowed

Property	Type	Description
<b>ports</b>	<b>array</b>	List of ports which should be made accessible on the pods selected for this rule. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.
<b>ports[]</b>	<b>object</b>	NetworkPolicyPort describes a port to allow traffic on

### 16.1.17. .spec.ingress[].from

#### Description

List of sources which should be able to access the pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all sources (traffic not restricted by source). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the from list.

#### Type

**array**

### 16.1.18. .spec.ingress[].from[]

#### Description

NetworkPolicyPeer describes a peer to allow traffic from. Only certain combinations of fields are allowed

#### Type

**object**

Property	Type	Description
<b>ipBlock</b>	<b>object</b>	IPBlock defines policy on a particular IPBlock. If this field is set then neither of the other fields can be.

Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	Selects Namespaces using cluster-scoped labels. This field follows standard label selector semantics; if present but empty, it selects all namespaces. If PodSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects all Pods in the Namespaces selected by NamespaceSelector.
<b>podSelector</b>	<b>object</b>	This is a label selector which selects Pods. This field follows standard label selector semantics; if present but empty, it selects all pods. If NamespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the Pods matching PodSelector in the policy's own Namespace.

### 16.1.19. .spec.ingress[].from[].ipBlock

#### Description

IPBlock defines policy on a particular IPBlock. If this field is set then neither of the other fields can be.

#### Type

**object**

#### Required

- **cidr**

Property	Type	Description
<b>cidr</b>	<b>string</b>	CIDR is a string representing the IP Block Valid examples are '192.168.1.1/24'

Property	Type	Description
<b>except</b>	<b>array (string)</b>	Except is a slice of CIDRs that should not be included within an IP Block Valid examples are '192.168.1.1/24' Except values will be rejected if they are outside the CIDR range

### 16.1.20. .spec.ingress[].from[].namespaceSelector

#### Description

Selects Namespaces using cluster-scoped labels. This field follows standard label selector semantics; if present but empty, it selects all namespaces. If PodSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects all Pods in the Namespaces selected by NamespaceSelector.

#### Type

**object**

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

### 16.1.21. .spec.ingress[].from[].namespaceSelector.matchExpressions

#### Description

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

#### Type

**array**



### 16.1.22. .spec.ingress[].from[].namespaceSelector.matchExpressions[]

#### Description

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

#### Type

**object**

#### Required

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

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

### 16.1.23. .spec.ingress[].from[].podSelector

#### Description

This is a label selector which selects Pods. This field follows standard label selector semantics; if present but empty, it selects all pods. If NamespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the Pods matching PodSelector in the policy's own Namespace.

#### Type

**object**

Property	Type	Description
<b>matchExpressions</b>	<b>array</b>	matchExpressions is a list of label selector requirements. The requirements are ANDed.

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

#### 16.1.24. .spec.ingress[].from[].podSelector.matchExpressions

##### Description

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

##### Type

**array**

#### 16.1.25. .spec.ingress[].from[].podSelector.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

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

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

### 16.1.26. .spec.ingress[].ports

#### Description

List of ports which should be made accessible on the pods selected for this rule. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.

#### Type

**array**

### 16.1.27. .spec.ingress[].ports[]

#### Description

NetworkPolicyPort describes a port to allow traffic on

#### Type

**object**

Property	Type	Description
<b>port</b>	<b>integer-or-string</b>	The port on the given protocol. This can either be a numerical or named port on a pod. If this field is not provided, this matches all port names and numbers.
<b>protocol</b>	<b>string</b>	The protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.

### 16.1.28. .spec.podSelector

#### Description

This is a label selector which selects Pods. This field follows standard label selector semantics; if present but empty, it selects all pods. If NamespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the Pods matching PodSelector in the Namespaces selected by

NamespaceSelector. Otherwise it selects the Pods matching PodSelector in the policy's own Namespace.

#### Type

**object**

Property	Type	Description
<b>matchExpressions</b>	<b>array</b>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<b>matchExpressions[]</b>	<b>object</b>	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
<b>matchLabels</b>	<b>object (string)</b>	

#### 16.1.29. .spec.podSelector.matchExpressions

##### Description

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

##### Type

**array**

#### 16.1.30. .spec.podSelector.matchExpressions[]

##### Description

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

##### Type

**object**

##### Required

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

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

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

## 16.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/k8s.cni.cncf.io/v1beta1/multi-networkpolicies**
  - **GET**: list objects of kind MultiNetworkPolicy
- **/apis/k8s.cni.cncf.io/v1beta1/namespaces/{namespace}/multi-networkpolicies**
  - **DELETE**: delete collection of MultiNetworkPolicy
  - **GET**: list objects of kind MultiNetworkPolicy
  - **POST**: create a MultiNetworkPolicy
- **/apis/k8s.cni.cncf.io/v1beta1/namespaces/{namespace}/multi-networkpolicies/{name}**
  - **DELETE**: delete a MultiNetworkPolicy
  - **GET**: read the specified MultiNetworkPolicy
  - **PATCH**: partially update the specified MultiNetworkPolicy
  - **PUT**: replace the specified MultiNetworkPolicy

### 16.2.1. /apis/k8s.cni.cncf.io/v1beta1/multi-networkpolicies

HTTP method

**GET**

Description

list objects of kind MultiNetworkPolicy

Table 16.1. HTTP responses

HTTP code	Reponse body
200 - OK	<b>MultiNetworkPolicyList</b> schema
401 - Unauthorized	Empty

## 16.2.2. /apis/k8s.cni.cncf.io/v1beta1/namespaces/{namespace}/multi-networkpolicies

HTTP method

**DELETE**

Description

delete collection of MultiNetworkPolicy

Table 16.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

HTTP method

**GET**

Description

list objects of kind MultiNetworkPolicy

Table 16.3. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">MultiNetworkPolicyList</a> schema
401 - Unauthorized	Empty

HTTP method

**POST**

Description

create a MultiNetworkPolicy

Table 16.4. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 16.5. Body parameters

Parameter	Type	Description
<b>body</b>	<b>MultiNetworkPolicy</b> schema	

Table 16.6. HTTP responses

HTTP code	Reponse body
200 - OK	<b>MultiNetworkPolicy</b> schema
201 - Created	<b>MultiNetworkPolicy</b> schema
202 - Accepted	<b>MultiNetworkPolicy</b> schema
401 - Unauthorized	Empty

### 16.2.3. /apis/k8s.cni.cncf.io/v1beta1/namespaces/{namespace}/multi-networkpolicies/{name}

Table 16.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the MultiNetworkPolicy

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

delete a MultiNetworkPolicy

**Table 16.8. Query parameters**

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

**Table 16.9. HTTP responses**

HTTP code	Response body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

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

read the specified MultiNetworkPolicy

**Table 16.10. HTTP responses**

HTTP code	Response body
200 - OK	<b>MultiNetworkPolicy</b> schema
401 - Unauthorized	Empty

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

partially update the specified MultiNetworkPolicy



Table 16.11. Query parameters

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

Table 16.12. HTTP responses

HTTP code	Response body
200 - OK	<b>MultiNetworkPolicy</b> schema
401 - Unauthorized	Empty

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

replace the specified MultiNetworkPolicy

Table 16.13. Query parameters

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

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

Table 16.14. Body parameters

Parameter	Type	Description
<b>body</b>	<b>MultiNetworkPolicy</b> schema	

Table 16.15. HTTP responses

HTTP code	Response body
200 - OK	<b>MultiNetworkPolicy</b> schema
201 - Created	<b>MultiNetworkPolicy</b> schema
401 - Unauthorized	Empty

## CHAPTER 17. NETWORKATTACHMENTDEFINITION [K8S.CNI.CNCF.IO/V1]

### Description

NetworkAttachmentDefinition is a CRD schema specified by the Network Plumbing Working Group to express the intent for attaching pods to one or more logical or physical networks. More information available at: <https://github.com/k8snetworkplumbingwg/multi-net-spec>

### Type

**object**

## 17.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	NetworkAttachmentDefinition spec defines the desired state of a network attachment

### 17.1.1. .spec

**Description**

NetworkAttachmentDefinition spec defines the desired state of a network attachment

**Type**

**object**

Property	Type	Description
<b>config</b>	<b>string</b>	NetworkAttachmentDefinition config is a JSON-formatted CNI configuration

## 17.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/k8s.cni.cncf.io/v1/network-attachment-definitions**
  - **GET**: list objects of kind NetworkAttachmentDefinition
- **/apis/k8s.cni.cncf.io/v1/namespaces/{namespace}/network-attachment-definitions**
  - **DELETE**: delete collection of NetworkAttachmentDefinition
  - **GET**: list objects of kind NetworkAttachmentDefinition
  - **POST**: create a NetworkAttachmentDefinition
- **/apis/k8s.cni.cncf.io/v1/namespaces/{namespace}/network-attachment-definitions/{name}**
  - **DELETE**: delete a NetworkAttachmentDefinition
  - **GET**: read the specified NetworkAttachmentDefinition
  - **PATCH**: partially update the specified NetworkAttachmentDefinition
  - **PUT**: replace the specified NetworkAttachmentDefinition

### 17.2.1. /apis/k8s.cni.cncf.io/v1/network-attachment-definitions

**HTTP method**

**GET**

**Description**

list objects of kind NetworkAttachmentDefinition

**Table 17.1. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">NetworkAttachmentDefinitionList</a> schema
401 - Unauthorized	Empty

## 17.2.2. /apis/k8s.cni.cncf.io/v1/namespaces/{namespace}/network-attachment-definitions

HTTP method

**DELETE**

Description

delete collection of NetworkAttachmentDefinition

Table 17.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

HTTP method

**GET**

Description

list objects of kind NetworkAttachmentDefinition

Table 17.3. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">NetworkAttachmentDefinitionList</a> schema
401 - Unauthorized	Empty

HTTP method

**POST**

Description

create a NetworkAttachmentDefinition

Table 17.4. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 17.5. Body parameters

Parameter	Type	Description
<b>body</b>	<b>NetworkAttachmentDefinition</b> schema	

Table 17.6. HTTP responses

HTTP code	Response body
200 - OK	<b>NetworkAttachmentDefinition</b> schema
201 - Created	<b>NetworkAttachmentDefinition</b> schema
202 - Accepted	<b>NetworkAttachmentDefinition</b> schema
401 - Unauthorized	Empty

### 17.2.3. /apis/k8s.cni.cncf.io/v1/namespaces/{namespace}/network-attachment-definitions/{name}

Table 17.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the NetworkAttachmentDefinition

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

delete a NetworkAttachmentDefinition

**Table 17.8. Query parameters**

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

**Table 17.9. HTTP responses**

HTTP code	Response body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

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

read the specified NetworkAttachmentDefinition

**Table 17.10. HTTP responses**

HTTP code	Response body
200 - OK	<b>NetworkAttachmentDefinition</b> schema
401 - Unauthorized	Empty

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

partially update the specified NetworkAttachmentDefinition

Table 17.11. Query parameters

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

Table 17.12. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">NetworkAttachmentDefinition</a> schema
401 - Unauthorized	Empty

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

replace the specified NetworkAttachmentDefinition

Table 17.13. Query parameters

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



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

Table 17.14. Body parameters

Parameter	Type	Description
<b>body</b>	<b>NetworkAttachmentDefinition</b> schema	

Table 17.15. HTTP responses

HTTP code	Response body
200 - OK	<b>NetworkAttachmentDefinition</b> schema
201 - Created	<b>NetworkAttachmentDefinition</b> schema
401 - Unauthorized	Empty

## CHAPTER 18. NETWORKPOLICY [NETWORKING.K8S.IO/V1]

### Description

NetworkPolicy describes what network traffic is allowed for a set of Pods

### Type

**object**

## 18.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	NetworkPolicySpec provides the specification of a NetworkPolicy

### 18.1.1. .spec

#### Description

NetworkPolicySpec provides the specification of a NetworkPolicy

#### Type

**object**

## Required

- **podSelector**

Property	Type	Description
<b>egress</b>	<b>array</b>	egress is a list of egress rules to be applied to the selected pods. Outgoing traffic is allowed if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic matches at least one egress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy limits all outgoing traffic (and serves solely to ensure that the pods it selects are isolated by default). This field is beta-level in 1.8
<b>egress[]</b>	<b>object</b>	NetworkPolicyEgressRule describes a particular set of traffic that is allowed out of pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and to. This type is beta-level in 1.8
<b>ingress</b>	<b>array</b>	ingress is a list of ingress rules to be applied to the selected pods. Traffic is allowed to a pod if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic source is the pod's local node, OR if the traffic matches at least one ingress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy does not allow any traffic (and serves solely to ensure that the pods it selects are isolated by default)

Property	Type	Description
<b>ingress[]</b>	<b>object</b>	NetworkPolicyIngressRule describes a particular set of traffic that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and from.
<b>podSelector</b>	<b>LabelSelector</b>	podSelector selects the pods to which this NetworkPolicy object applies. The array of ingress rules is applied to any pods selected by this field. Multiple network policies can select the same set of pods. In this case, the ingress rules for each are combined additively. This field is NOT optional and follows standard label selector semantics. An empty podSelector matches all pods in this namespace.
<b>policyTypes</b>	<b>array (string)</b>	policyTypes is a list of rule types that the NetworkPolicy relates to. Valid options are ["Ingress"], ["Egress"], or ["Ingress", "Egress"]. If this field is not specified, it will default based on the existence of ingress or egress rules; policies that contain an egress section are assumed to affect egress, and all policies (whether or not they contain an ingress section) are assumed to affect ingress. If you want to write an egress-only policy, you must explicitly specify policyTypes [ "Egress" ]. Likewise, if you want to write a policy that specifies that no egress is allowed, you must specify a policyTypes value that include "Egress" (since such a policy would not include an egress section and would otherwise default to just [ "Ingress" ]). This field is beta-level in 1.8

### 18.1.2. .spec.egress

#### Description

egress is a list of egress rules to be applied to the selected pods. Outgoing traffic is allowed if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic matches at least one egress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy limits all outgoing traffic (and serves solely to ensure that the pods it selects are isolated by default). This field is beta-level in 1.8

#### Type

**array**

### 18.1.3. .spec.egress[]

#### Description

NetworkPolicyEgressRule describes a particular set of traffic that is allowed out of pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and to. This type is beta-level in 1.8

#### Type

**object**

Property	Type	Description
<b>ports</b>	<b>array</b>	ports is a list of destination ports for outgoing traffic. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.
<b>ports[]</b>	<b>object</b>	NetworkPolicyPort describes a port to allow traffic on
<b>to</b>	<b>array</b>	to is a list of destinations for outgoing traffic of pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all destinations (traffic not restricted by destination). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the to list.
<b>to[]</b>	<b>object</b>	NetworkPolicyPeer describes a peer to allow traffic to/from. Only certain combinations of fields are allowed

### 18.1.4. .spec.egress[].ports

#### Description

ports is a list of destination ports for outgoing traffic. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.

#### Type

**array**

### 18.1.5. .spec.egress[].ports[]

#### Description

NetworkPolicyPort describes a port to allow traffic on

#### Type

**object**

Property	Type	Description
<b>endPort</b>	<b>integer</b>	endPort indicates that the range of ports from port to endPort if set, inclusive, should be allowed by the policy. This field cannot be defined if the port field is not defined or if the port field is defined as a named (string) port. The endPort must be equal or greater than port.
<b>port</b>	<b>IntOrString</b>	port represents the port on the given protocol. This can either be a numerical or named port on a pod. If this field is not provided, this matches all port names and numbers. If present, only traffic on the specified protocol AND port will be matched.
<b>protocol</b>	<b>string</b>	protocol represents the protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.  Possible enum values: - <b>"SCTP"</b> is the SCTP protocol. - <b>"TCP"</b> is the TCP protocol. - <b>"UDP"</b> is the UDP protocol.

### 18.1.6. .spec.egress[].to

**Description**

to is a list of destinations for outgoing traffic of pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all destinations (traffic not restricted by destination). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the to list.

**Type**

**array**

**18.1.7. .spec.egress[].to[]****Description**

NetworkPolicyPeer describes a peer to allow traffic to/from. Only certain combinations of fields are allowed

**Type**

**object**

Property	Type	Description
<b>ipBlock</b>	<b>object</b>	IPBlock describes a particular CIDR (Ex. "192.168.1.0/24","2001:db8::/64") that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The except entry describes CIDRs that should not be included within this rule.
<b>namespaceSelector</b>	<b>LabelSelector</b>	<p>namespaceSelector selects namespaces using cluster-scoped labels. This field follows standard label selector semantics; if present but empty, it selects all namespaces.</p> <p>If podSelector is also set, then the NetworkPolicyPeer as a whole selects the pods matching podSelector in the namespaces selected by namespaceSelector. Otherwise it selects all pods in the namespaces selected by namespaceSelector.</p>

Property	Type	Description
<b>podSelector</b>	<b>LabelSelector</b>	<p>podSelector is a label selector which selects pods. This field follows standard label selector semantics; if present but empty, it selects all pods.</p> <p>If namespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the pods matching podSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the pods matching podSelector in the policy's own namespace.</p>

### 18.1.8. .spec.egress[].to[].ipBlock

#### Description

IPBlock describes a particular CIDR (Ex. "192.168.1.0/24","2001:db8::/64") that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The except entry describes CIDRs that should not be included within this rule.

#### Type

**object**

#### Required

- **cidr**

Property	Type	Description
<b>cidr</b>	<b>string</b>	<p>cidr is a string representing the IPBlock Valid examples are "192.168.1.0/24" or "2001:db8::/64"</p>
<b>except</b>	<b>array (string)</b>	<p>except is a slice of CIDRs that should not be included within an IPBlock Valid examples are "192.168.1.0/24" or "2001:db8::/64" Except values will be rejected if they are outside the cidr range</p>

### 18.1.9. .spec.ingress

#### Description

ingress is a list of ingress rules to be applied to the selected pods. Traffic is allowed to a pod if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the



traffic source is the pod's local node, OR if the traffic matches at least one ingress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy does not allow any traffic (and serves solely to ensure that the pods it selects are isolated by default)

#### Type

**array**

### 18.1.10. .spec.ingress[]

#### Description

NetworkPolicyIngressRule describes a particular set of traffic that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and from.

#### Type

**object**

Property	Type	Description
<b>from</b>	<b>array</b>	from is a list of sources which should be able to access the pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all sources (traffic not restricted by source). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the from list.
<b>from[]</b>	<b>object</b>	NetworkPolicyPeer describes a peer to allow traffic to/from. Only certain combinations of fields are allowed
<b>ports</b>	<b>array</b>	ports is a list of ports which should be made accessible on the pods selected for this rule. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.
<b>ports[]</b>	<b>object</b>	NetworkPolicyPort describes a port to allow traffic on

### 18.1.11. .spec.ingress[].from

#### Description

from is a list of sources which should be able to access the pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all sources (traffic not restricted by source). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the from list.

#### Type

**array**

### 18.1.12. .spec.ingress[].from[]

#### Description

NetworkPolicyPeer describes a peer to allow traffic to/from. Only certain combinations of fields are allowed

#### Type

**object**

Property	Type	Description
<b>ipBlock</b>	<b>object</b>	IPBlock describes a particular CIDR (Ex. "192.168.1.0/24", "2001:db8::/64") that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The except entry describes CIDRs that should not be included within this rule.
<b>namespaceSelector</b>	<b>LabelSelector</b>	<p>namespaceSelector selects namespaces using cluster-scoped labels. This field follows standard label selector semantics; if present but empty, it selects all namespaces.</p> <p>If podSelector is also set, then the NetworkPolicyPeer as a whole selects the pods matching podSelector in the namespaces selected by namespaceSelector. Otherwise it selects all pods in the namespaces selected by namespaceSelector.</p>

Property	Type	Description
<b>podSelector</b>	<b>LabelSelector</b>	<p>podSelector is a label selector which selects pods. This field follows standard label selector semantics; if present but empty, it selects all pods.</p> <p>If namespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the pods matching podSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the pods matching podSelector in the policy's own namespace.</p>

### 18.1.13. .spec.ingress[].from[].ipBlock

#### Description

IPBlock describes a particular CIDR (Ex. "192.168.1.0/24", "2001:db8::/64") that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The except entry describes CIDRs that should not be included within this rule.

#### Type

**object**

#### Required

- **cidr**

Property	Type	Description
<b>cidr</b>	<b>string</b>	<p>cidr is a string representing the IPBlock Valid examples are "192.168.1.0/24" or "2001:db8::/64"</p>
<b>except</b>	<b>array (string)</b>	<p>except is a slice of CIDRs that should not be included within an IPBlock Valid examples are "192.168.1.0/24" or "2001:db8::/64" Except values will be rejected if they are outside the cidr range</p>

### 18.1.14. .spec.ingress[].ports

#### Description

ports is a list of ports which should be made accessible on the pods selected for this rule. Each item in

this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.

#### Type

**array**

### 18.1.15. .spec.ingress[].ports[]

#### Description

NetworkPolicyPort describes a port to allow traffic on

#### Type

**object**

Property	Type	Description
<b>endPort</b>	<b>integer</b>	endPort indicates that the range of ports from port to endPort if set, inclusive, should be allowed by the policy. This field cannot be defined if the port field is not defined or if the port field is defined as a named (string) port. The endPort must be equal or greater than port.
<b>port</b>	<b>IntOrString</b>	port represents the port on the given protocol. This can either be a numerical or named port on a pod. If this field is not provided, this matches all port names and numbers. If present, only traffic on the specified protocol AND port will be matched.
<b>protocol</b>	<b>string</b>	<p>protocol represents the protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.</p> <p>Possible enum values: - <b>"SCTP"</b> is the SCTP protocol. - <b>"TCP"</b> is the TCP protocol. - <b>"UDP"</b> is the UDP protocol.</p>

## 18.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/networking.k8s.io/v1/networkpolicies**

- **GET**: list or watch objects of kind NetworkPolicy
- **/apis/networking.k8s.io/v1/watch/networkpolicies**
  - **GET**: watch individual changes to a list of NetworkPolicy. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies**
  - **DELETE**: delete collection of NetworkPolicy
  - **GET**: list or watch objects of kind NetworkPolicy
  - **POST**: create a NetworkPolicy
- **/apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies**
  - **GET**: watch individual changes to a list of NetworkPolicy. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}**
  - **DELETE**: delete a NetworkPolicy
  - **GET**: read the specified NetworkPolicy
  - **PATCH**: partially update the specified NetworkPolicy
  - **PUT**: replace the specified NetworkPolicy
- **/apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies/{name}**
  - **GET**: watch changes to an object of kind NetworkPolicy. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

### 18.2.1. /apis/networking.k8s.io/v1/networkpolicies

HTTP method

**GET**

Description

list or watch objects of kind NetworkPolicy

Table 18.1. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">NetworkPolicyList</a> schema
401 - Unauthorized	Empty

### 18.2.2. /apis/networking.k8s.io/v1/watch/networkpolicies

HTTP method

**GET****Description**

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

**Table 18.2. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

**18.2.3. /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies****HTTP method****DELETE****Description**

delete collection of NetworkPolicy

**Table 18.3. Query parameters**

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

**Table 18.4. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

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

list or watch objects of kind NetworkPolicy

**Table 18.5. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">NetworkPolicyList</a> schema
401 - Unauthorized	Empty

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

create a NetworkPolicy

**Table 18.6. Query parameters**

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

**Table 18.7. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">NetworkPolicy</a> schema	

**Table 18.8. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">NetworkPolicy</a> schema
201 - Created	<a href="#">NetworkPolicy</a> schema
202 - Accepted	<a href="#">NetworkPolicy</a> schema
401 - Unauthorized	Empty

#### 18.2.4. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies

##### HTTP method

##### GET

##### Description

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

Table 18.9. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

#### 18.2.5. /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}

Table 18.10. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the NetworkPolicy

##### HTTP method

##### DELETE

##### Description

delete a NetworkPolicy

Table 18.11. Query parameters

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



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

Table 18.12. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

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

read the specified NetworkPolicy

Table 18.13. HTTP responses

HTTP code	Reponse body
200 - OK	<b>NetworkPolicy</b> schema
401 - Unauthorized	Empty

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

partially update the specified NetworkPolicy

Table 18.14. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 18.15. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">NetworkPolicy</a> schema
201 - Created	<a href="#">NetworkPolicy</a> schema
401 - Unauthorized	Empty

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

replace the specified NetworkPolicy

Table 18.16. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 18.17. Body parameters

Parameter	Type	Description
<b>body</b>	<b>NetworkPolicy</b> schema	

Table 18.18. HTTP responses

HTTP code	Response body
200 - OK	<b>NetworkPolicy</b> schema
201 - Created	<b>NetworkPolicy</b> schema
401 - Unauthorized	Empty

### 18.2.6. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies/

Table 18.19. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the NetworkPolicy

HTTP method

## GET

### Description

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

Table 18.20. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

## CHAPTER 19. OVERLAPPINGRANGEIPRESERVATION [WHEREABOUTS.CNI.CNCF.IO/V1ALPHA1]

### Description

OverlappingRangeIPReservation is the Schema for the OverlappingRangeIPReservations API

### Type

**object**

### Required

- **spec**

## 19.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	OverlappingRangeIPReservationSpec defines the desired state of OverlappingRangeIPReservation

### 19.1.1. .spec

#### Description

OverlappingRangeIPReservationSpec defines the desired state of OverlappingRangeIPReservation

#### Type

**object**

#### Required

- **podref**

Property	Type	Description
<b>containerid</b>	<b>string</b>	
<b>ifname</b>	<b>string</b>	
<b>podref</b>	<b>string</b>	

## 19.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/whereabouts.cni.cncf.io/v1alpha1/overlappingrangeipreservations**
  - **GET**: list objects of kind OverlappingRangeIPReservation
- **/apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/overlappingrangeipreservations**
  - **DELETE**: delete collection of OverlappingRangeIPReservation
  - **GET**: list objects of kind OverlappingRangeIPReservation
  - **POST**: create an OverlappingRangeIPReservation
- **/apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/overlappingrangeipreservations/{name}**
  - **DELETE**: delete an OverlappingRangeIPReservation
  - **GET**: read the specified OverlappingRangeIPReservation
  - **PATCH**: partially update the specified OverlappingRangeIPReservation
  - **PUT**: replace the specified OverlappingRangeIPReservation

### 19.2.1. /apis/whereabouts.cni.cncf.io/v1alpha1/overlappingrangeipreservations

#### HTTP method

**GET**

#### Description

list objects of kind `OverlappingRangeIPReservation`

**Table 19.1. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">OverlappingRangeIPReservationList</a> schema
401 - Unauthorized	Empty

## 19.2.2. /apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/overlapping

HTTP method

**DELETE**

Description

delete collection of `OverlappingRangeIPReservation`

**Table 19.2. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

HTTP method

**GET**

Description

list objects of kind `OverlappingRangeIPReservation`

**Table 19.3. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">OverlappingRangeIPReservationList</a> schema
401 - Unauthorized	Empty

HTTP method

**POST**

Description

create an `OverlappingRangeIPReservation`

**Table 19.4. Query parameters**

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

Table 19.5. Body parameters

Parameter	Type	Description
<b>body</b>	<b>OverlappingRangeIPReservation</b> schema	

Table 19.6. HTTP responses

HTTP code	Response body
200 - OK	<b>OverlappingRangeIPReservation</b> schema
201 - Created	<b>OverlappingRangeIPReservation</b> schema
202 - Accepted	<b>OverlappingRangeIPReservation</b> schema
401 - Unauthorized	Empty

19.2.3. /apis/whereabouts.cni.cncf.io/v1alpha1/namespaces/{namespace}/overlapping



Table 19.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the OverlappingRangeIPReservation

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

delete an OverlappingRangeIPReservation

Table 19.8. Query parameters

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

Table 19.9. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

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

read the specified OverlappingRangeIPReservation

Table 19.10. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">OverlappingRangeIPReservation</a> schema
401 - Unauthorized	Empty

**HTTP method****PATCH**

**Description**

partially update the specified `OverlappingRangeIPReservation`

**Table 19.11. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<b>fieldValidation</b>	<b>string</b>	<code>fieldValidation</code> 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 <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

**Table 19.12. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">OverlappingRangeIPReservation</a> schema
401 - Unauthorized	Empty

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

replace the specified `OverlappingRangeIPReservation`

**Table 19.13. Query parameters**

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

Table 19.14. Body parameters

Parameter	Type	Description
<b>body</b>	<b>OverlappingRangeIPReservation</b> schema	

Table 19.15. HTTP responses

HTTP code	Response body
200 - OK	<b>OverlappingRangeIPReservation</b> schema
201 - Created	<b>OverlappingRangeIPReservation</b> schema
401 - Unauthorized	Empty

## CHAPTER 20. PODNETWORKCONNECTIVITYCHECK [CONTROLPLANE.OPERATOR.OPENSIFT.IO/V1ALPHA1]

### Description

PodNetworkConnectivityCheck

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

- **spec**

## 20.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	Spec defines the source and target of the connectivity check

Property	Type	Description
<b>status</b>	<b>object</b>	Status contains the observed status of the connectivity check

### 20.1.1. .spec

#### Description

Spec defines the source and target of the connectivity check

#### Type

**object**

#### Required

- **sourcePod**
- **targetEndpoint**

Property	Type	Description
<b>sourcePod</b>	<b>string</b>	SourcePod names the pod from which the condition will be checked
<b>targetEndpoint</b>	<b>string</b>	EndpointAddress to check. A TCP address of the form host:port. Note that if host is a DNS name, then the check would fail if the DNS name cannot be resolved. Specify an IP address for host to bypass DNS name lookup.
<b>tlsClientCert</b>	<b>object</b>	TLSCert, if specified, references a kubernetes.io/tls type secret with 'tls.crt' and 'tls.key' entries containing an optional TLS client certificate and key to be used when checking endpoints that require a client certificate in order to gracefully perform the scan without causing excessive logging in the endpoint process. The secret must exist in the same namespace as this resource.

### 20.1.2. .spec.tlsClientCert

#### Description

TLSCert, if specified, references a kubernetes.io/tls type secret with 'tls.crt' and 'tls.key' entries

containing an optional TLS client certificate and key to be used when checking endpoints that require a client certificate in order to gracefully preform the scan without causing excessive logging in the endpoint process. The secret must exist in the same namespace as this resource.

#### Type

**object**

#### Required

- **name**

Property	Type	Description
<b>name</b>	<b>string</b>	name is the metadata.name of the referenced secret

### 20.1.3. .status

#### Description

Status contains the observed status of the connectivity check

#### Type

**object**

Property	Type	Description
<b>conditions</b>	<b>array</b>	Conditions summarize the status of the check
<b>conditions[]</b>	<b>object</b>	PodNetworkConnectivityCheckCondition represents the overall status of the pod network connectivity.
<b>failures</b>	<b>array</b>	Failures contains logs of unsuccessful check actions
<b>failures[]</b>	<b>object</b>	LogEntry records events
<b>outages</b>	<b>array</b>	Outages contains logs of time periods of outages
<b>outages[]</b>	<b>object</b>	OutageEntry records time period of an outage
<b>successes</b>	<b>array</b>	Successes contains logs successful check actions
<b>successes[]</b>	<b>object</b>	LogEntry records events

### 20.1.4. .status.conditions

#### Description

Conditions summarize the status of the check

#### Type

**array**

### 20.1.5. .status.conditions[]

#### Description

PodNetworkConnectivityCheckCondition represents the overall status of the pod network connectivity.

#### Type

**object**

#### Required

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

Property	Type	Description
<b>lastTransitionTime</b>	``	Last time the condition transitioned from one status to another.
<b>message</b>	<b>string</b>	Message indicating details about last transition in a human readable format.
<b>reason</b>	<b>string</b>	Reason for the condition's last status transition in a machine readable format.
<b>status</b>	<b>string</b>	Status of the condition
<b>type</b>	<b>string</b>	Type of the condition

### 20.1.6. .status.failures

#### Description

Failures contains logs of unsuccessful check actions

#### Type

**array**

### 20.1.7. .status.failures[]

#### Description

LogEntry records events

Type

**object**

Required

- **success**

Property	Type	Description
<b>latency</b>	``	Latency records how long the action mentioned in the entry took.
<b>message</b>	<b>string</b>	Message explaining status in a human readable format.
<b>reason</b>	<b>string</b>	Reason for status in a machine readable format.
<b>success</b>	<b>boolean</b>	Success indicates if the log entry indicates a success or failure.
<b>time</b>	``	Start time of check action.

### 20.1.8. .status.outages

Description

Outages contains logs of time periods of outages

Type

**array**

### 20.1.9. .status.outages[]

Description

OutageEntry records time period of an outage

Type

**object**

Property	Type	Description
<b>end</b>	``	End of outage detected



Property	Type	Description
<b>endLogs</b>	<b>array</b>	EndLogs contains log entries related to the end of this outage. Should contain the success entry that resolved the outage and possibly a few of the failure log entries that preceded it.
<b>endLogs[]</b>	<b>object</b>	LogEntry records events
<b>message</b>	<b>string</b>	Message summarizes outage details in a human readable format.
<b>start</b>	<b>..</b>	Start of outage detected
<b>startLogs</b>	<b>array</b>	StartLogs contains log entries related to the start of this outage. Should contain the original failure, any entries where the failure mode changed.
<b>startLogs[]</b>	<b>object</b>	LogEntry records events

### 20.1.10. .status.outages[].endLogs

#### Description

EndLogs contains log entries related to the end of this outage. Should contain the success entry that resolved the outage and possibly a few of the failure log entries that preceded it.

#### Type

**array**

### 20.1.11. .status.outages[].endLogs[]

#### Description

LogEntry records events

#### Type

**object**

#### Required

- **success**

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

Property	Type	Description
<b>latency</b>	``	Latency records how long the action mentioned in the entry took.
<b>message</b>	<b>string</b>	Message explaining status in a human readable format.
<b>reason</b>	<b>string</b>	Reason for status in a machine readable format.
<b>success</b>	<b>boolean</b>	Success indicates if the log entry indicates a success or failure.
<b>time</b>	``	Start time of check action.

### 20.1.12. `.status.outages[].startLogs`

#### Description

StartLogs contains log entries related to the start of this outage. Should contain the original failure, any entries where the failure mode changed.

#### Type

**array**

### 20.1.13. `.status.outages[].startLogs[]`

#### Description

LogEntry records events

#### Type

**object**

#### Required

- **success**

Property	Type	Description
<b>latency</b>	``	Latency records how long the action mentioned in the entry took.
<b>message</b>	<b>string</b>	Message explaining status in a human readable format.
<b>reason</b>	<b>string</b>	Reason for status in a machine readable format.

Property	Type	Description
<b>success</b>	<b>boolean</b>	Success indicates if the log entry indicates a success or failure.
<b>time</b>	..	Start time of check action.

#### 20.1.14. .status.successes

##### Description

Successes contains logs successful check actions

##### Type

**array**

#### 20.1.15. .status.successes[]

##### Description

LogEntry records events

##### Type

**object**

##### Required

- **success**

Property	Type	Description
<b>latency</b>	..	Latency records how long the action mentioned in the entry took.
<b>message</b>	<b>string</b>	Message explaining status in a human readable format.
<b>reason</b>	<b>string</b>	Reason for status in a machine readable format.
<b>success</b>	<b>boolean</b>	Success indicates if the log entry indicates a success or failure.
<b>time</b>	..	Start time of check action.

## 20.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/controlplane.operator.openshift.io/v1alpha1/podnetworkconnectivitychecks**

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

### 20.2.1. /apis/controlplane.operator.openshift.io/v1alpha1/podnetworkconnectivityche

#### HTTP method

#### GET

#### Description

list objects of kind PodNetworkConnectivityCheck

Table 20.1. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">PodNetworkConnectivityCheckList</a> schema
401 - Unauthorized	Empty

### 20.2.2. /apis/controlplane.operator.openshift.io/v1alpha1/namespaces/{namespace}/

#### HTTP method

#### DELETE

#### Description

delete collection of PodNetworkConnectivityCheck

Table 20.2. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
401 - Unauthorized	Empty

## HTTP method

**GET**

## Description

list objects of kind PodNetworkConnectivityCheck

Table 20.3. HTTP responses

HTTP code	Reponse body
200 - OK	<b>PodNetworkConnectivityCheckList</b> schema
401 - Unauthorized	Empty

## HTTP method

**POST**

## Description

create a PodNetworkConnectivityCheck

Table 20.4. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 20.5. Body parameters

Parameter	Type	Description
<b>body</b>	<b>PodNetworkConnectivityCheck</b> schema	

Table 20.6. HTTP responses

HTTP code	Response body
200 - OK	<b>PodNetworkConnectivityCheck</b> schema
201 - Created	<b>PodNetworkConnectivityCheck</b> schema
202 - Accepted	<b>PodNetworkConnectivityCheck</b> schema
401 - Unauthorized	Empty

### 20.2.3. /apis/controlplane.operator.openshift.io/v1alpha1/namespaces/{namespace}/

Table 20.7. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the PodNetworkConnectivityCheck

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

delete a PodNetworkConnectivityCheck

**Table 20.8. Query parameters**

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

**Table 20.9. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

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

read the specified PodNetworkConnectivityCheck

**Table 20.10. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">PodNetworkConnectivityCheck</a> schema
401 - Unauthorized	Empty

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

partially update the specified PodNetworkConnectivityCheck

**Table 20.11. Query parameters**

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

Table 20.12. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">PodNetworkConnectivityCheck</a> schema
401 - Unauthorized	Empty

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

replace the specified PodNetworkConnectivityCheck

Table 20.13. Query parameters

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



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

Table 20.14. Body parameters

Parameter	Type	Description
<b>body</b>	<b>PodNetworkConnectivityCheck</b> schema	

Table 20.15. HTTP responses

HTTP code	Response body
200 - OK	<b>PodNetworkConnectivityCheck</b> schema
201 - Created	<b>PodNetworkConnectivityCheck</b> schema
401 - Unauthorized	Empty

#### 20.2.4. /apis/controlplane.operator.openshift.io/v1alpha1/namespaces/{namespace}/

Table 20.16. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the PodNetworkConnectivityCheck

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

read status of the specified PodNetworkConnectivityCheck

**Table 20.17. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">PodNetworkConnectivityCheck</a> schema
401 - Unauthorized	Empty

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

partially update status of the specified PodNetworkConnectivityCheck

**Table 20.18. Query parameters**

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

Table 20.19. HTTP responses

HTTP code	Response body
200 - OK	<b>PodNetworkConnectivityCheck</b> schema
401 - Unauthorized	Empty

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

replace status of the specified PodNetworkConnectivityCheck

Table 20.20. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 20.21. Body parameters

Parameter	Type	Description
<b>body</b>	<b>PodNetworkConnectivityCheck</b> schema	

Table 20.22. HTTP responses

HTTP code	Response body
200 - OK	<b>PodNetworkConnectivityCheck</b> schema
201 - Created	<b>PodNetworkConnectivityCheck</b> schema
401 - Unauthorized	Empty

## CHAPTER 21. ROUTE [ROUTE.OPENSIFT.IO/V1]

### Description

A route allows developers to expose services through an HTTP(S) aware load balancing and proxy layer via a public DNS entry. The route may further specify TLS options and a certificate, or specify a public CNAME that the router should also accept for HTTP and HTTPS traffic. An administrator typically configures their router to be visible outside the cluster firewall, and may also add additional security, caching, or traffic controls on the service content. Routers usually talk directly to the service endpoints.

Once a route is created, the **host** field may not be changed. Generally, routers use the oldest route with a given host when resolving conflicts.

Routers are subject to additional customization and may support additional controls via the annotations field.

Because administrators may configure multiple routers, the route status field is used to return information to clients about the names and states of the route under each router. If a client chooses a duplicate name, for instance, the route status conditions are used to indicate the route cannot be chosen.

To enable HTTP/2 ALPN on a route it requires a custom (non-wildcard) certificate. This prevents connection coalescing by clients, notably web browsers. We do not support HTTP/2 ALPN on routes that use the default certificate because of the risk of connection re-use/coalescing. Routes that do not have their own custom certificate will not be HTTP/2 ALPN-enabled on either the frontend or the backend.

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

### Type

**object**

### Required

- **spec**

## 21.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>

Property	Type	Description
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b><a href="#">ObjectMeta_v2</a></b>	metadata is the standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>

Property	Type	Description
<b>spec</b>	<b>object</b>	<p>RouteSpec describes the hostname or path the route exposes, any security information, and one to four backends (services) the route points to. Requests are distributed among the backends depending on the weights assigned to each backend. When using roundrobin scheduling the portion of requests that go to each backend is the backend weight divided by the sum of all of the backend weights. When the backend has more than one endpoint the requests that end up on the backend are roundrobin distributed among the endpoints. Weights are between 0 and 256 with default 100. Weight 0 causes no requests to the backend. If all weights are zero the route will be considered to have no backends and return a standard 503 response.</p> <p>The <b>tls</b> field is optional and allows specific certificates or behavior for the route. Routers typically configure a default certificate on a wildcard domain to terminate routes without explicit certificates, but custom hostnames usually must choose passthrough (send traffic directly to the backend via the TLS Server-Name- Indication field) or provide a certificate.</p>
<b>status</b>	<b>object</b>	RouteStatus provides relevant info about the status of a route, including which routers acknowledge it.

### 21.1.1. .spec

#### Description

RouteSpec describes the hostname or path the route exposes, any security information, and one to four backends (services) the route points to. Requests are distributed among the backends depending on the weights assigned to each backend. When using roundrobin scheduling the portion of requests that go to each backend is the backend weight divided by the sum of all of the backend

weights. When the backend has more than one endpoint the requests that end up on the backend are roundrobin distributed among the endpoints. Weights are between 0 and 256 with default 100. Weight 0 causes no requests to the backend. If all weights are zero the route will be considered to have no backends and return a standard 503 response.

The **tls** field is optional and allows specific certificates or behavior for the route. Routers typically configure a default certificate on a wildcard domain to terminate routes without explicit certificates, but custom hostnames usually must choose passthrough (send traffic directly to the backend via the TLS Server-Name- Indication field) or provide a certificate.

## Type

**object**

## Required

- **to**

Property	Type	Description
<b>alternateBackends</b>	<b>array</b>	alternateBackends allows up to 3 additional backends to be assigned to the route. Only the Service kind is allowed, and it will be defaulted to Service. Use the weight field in RouteTargetReference object to specify relative preference.
<b>alternateBackends[]</b>	<b>object</b>	RouteTargetReference specifies the target that resolve into endpoints. Only the 'Service' kind is allowed. Use 'weight' field to emphasize one over others.
<b>host</b>	<b>string</b>	host is an alias/DNS that points to the service. Optional. If not specified a route name will typically be automatically chosen. Must follow DNS952 subdomain conventions.
<b>httpHeaders</b>	<b>object</b>	RouteHTTPHeaders defines policy for HTTP headers.
<b>path</b>	<b>string</b>	path that the router watches for, to route traffic for to the service. Optional
<b>port</b>	<b>object</b>	RoutePort defines a port mapping from a router to an endpoint in the service endpoints.



Property	Type	Description
<b>subdomain</b>	<b>string</b>	<p>subdomain is a DNS subdomain that is requested within the ingress controller's domain (as a subdomain). If host is set this field is ignored. An ingress controller may choose to ignore this suggested name, in which case the controller will report the assigned name in the status.ingress array or refuse to admit the route. If this value is set and the server does not support this field host will be populated automatically. Otherwise host is left empty. The field may have multiple parts separated by a dot, but not all ingress controllers may honor the request. This field may not be changed after creation except by a user with the update routes/custom-host permission.</p> <p>Example: subdomain <b>frontend</b> automatically receives the router subdomain <b>apps.mycluster.com</b> to have a full hostname <b>frontend.apps.mycluster.com</b>.</p>
<b>tls</b>	<b>object</b>	<p>TLSConfig defines config used to secure a route and provide termination</p>
<b>to</b>	<b>object</b>	<p>RouteTargetReference specifies the target that resolve into endpoints. Only the 'Service' kind is allowed. Use 'weight' field to emphasize one over others.</p>
<b>wildcardPolicy</b>	<b>string</b>	<p>Wildcard policy if any for the route. Currently only 'Subdomain' or 'None' is allowed.</p>

## 21.1.2. .spec.alternateBackends

### Description

alternateBackends allows up to 3 additional backends to be assigned to the route. Only the Service kind is allowed, and it will be defaulted to Service. Use the weight field in RouteTargetReference object to specify relative preference.

**Type****array**

### 21.1.3. .spec.alternateBackends[]

**Description**

RouteTargetReference specifies the target that resolve into endpoints. Only the 'Service' kind is allowed. Use 'weight' field to emphasize one over others.

**Type****object****Required**

- **kind**
- **name**

Property	Type	Description
<b>kind</b>	<b>string</b>	The kind of target that the route is referring to. Currently, only 'Service' is allowed
<b>name</b>	<b>string</b>	name of the service/target that is being referred to. e.g. name of the service
<b>weight</b>	<b>integer</b>	weight as an integer between 0 and 256, default 100, that specifies the target's relative weight against other target reference objects. 0 suppresses requests to this backend.

### 21.1.4. .spec.httpHeaders

**Description**

RouteHTTPHeaders defines policy for HTTP headers.

**Type****object**

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

Property	Type	Description
<b>actions</b>	<b>object</b>	RouteHTTPHeaderActions defines configuration for actions on HTTP request and response headers.

### 21.1.5. .spec.httpHeaders.actions

#### Description

RouteHTTPHeaderActions defines configuration for actions on HTTP request and response headers.

#### Type

**object**

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

Property	Type	Description
<b>request</b>	<b>array</b>	request is a list of HTTP request headers to modify. Currently, actions may define to either <b>Set</b> or <b>Delete</b> headers values. Actions defined here will modify the request headers of all requests made through a route. These actions are applied to a specific Route defined within a cluster i.e. connections made through a route. Currently, actions may define to either <b>Set</b> or <b>Delete</b> headers values. Route actions will be executed after IngressController actions for request headers. Actions are applied in sequence as defined in this list. A maximum of 20 request header actions may be configured. You can use this field to specify HTTP request headers that should be set or deleted when forwarding connections from the client to your application. Sample fetchers allowed are "req.hdr" and "ssl_c_der". Converters allowed are "lower" and "base64". Example header values: "[%req.hdr(X-target),lower]", "%{+Q}[ssl_c_der,base64]". Any request header configuration applied directly via a Route resource using this API will override header configuration for a header of the same name applied via spec.httpHeaders.actions on the IngressController or route annotation. Note: This field cannot be used if your route uses TLS passthrough.
<b>request[]</b>	<b>object</b>	RouteHTTPHeader specifies configuration for setting or deleting an HTTP header.

Property	Type	Description
<b>response</b>	<b>array</b>	response is a list of HTTP response headers to modify. Currently, actions may define to either <b>Set</b> or <b>Delete</b> headers values. Actions defined here will modify the response headers of all requests made through a route. These actions are applied to a specific Route defined within a cluster i.e. connections made through a route. Route actions will be executed before IngressController actions for response headers. Actions are applied in sequence as defined in this list. A maximum of 20 response header actions may be configured. You can use this field to specify HTTP response headers that should be set or deleted when forwarding responses from your application to the client. Sample fetchers allowed are "res.hdr" and "ssl_c_der". Converters allowed are "lower" and "base64". Example header values: "[%res.hdr(X-target),lower]", "%{+Q}[ssl_c_der,base64]". Note: This field cannot be used if your route uses TLS passthrough.
<b>response[]</b>	<b>object</b>	RouteHTTPHeader specifies configuration for setting or deleting an HTTP header.

### 21.1.6. .spec.httpHeaders.actions.request

#### Description

request is a list of HTTP request headers to modify. Currently, actions may define to either **Set** or **Delete** headers values. Actions defined here will modify the request headers of all requests made through a route. These actions are applied to a specific Route defined within a cluster i.e. connections made through a route. Currently, actions may define to either **Set** or **Delete** headers values. Route actions will be executed after IngressController actions for request headers. Actions are applied in sequence as defined in this list. A maximum of 20 request header actions may be configured. You can use this field to specify HTTP request headers that should be set or deleted when forwarding connections from the client to your application. Sample fetchers allowed are "req.hdr" and "ssl\_c\_der". Converters allowed are "lower" and "base64". Example header values: "[%req.hdr(X-target),lower]", "%{+Q}[ssl\_c\_der,base64]". Any request header configuration applied directly via a

Route resource using this API will override header configuration for a header of the same name applied via `spec.httpHeaders.actions` on the IngressController or route annotation. Note: This field cannot be used if your route uses TLS passthrough.

**Type****array****21.1.7. `.spec.httpHeaders.actions.request[]`****Description**

RouteHTTPHeader specifies configuration for setting or deleting an HTTP header.

**Type****object****Required**

- **name**
- **action**

Property	Type	Description
<b>action</b>	<b>object</b>	RouteHTTPHeaderActionUnion specifies an action to take on an HTTP header.
<b>name</b>	<b>string</b>	name specifies the name of a header on which to perform an action. Its value must be a valid HTTP header name as defined in RFC 2616 section 4.2. The name must consist only of alphanumeric and the following special characters, "-!#\$%&'*+.^_`". The following header names are reserved and may not be modified via this API: Strict-Transport-Security, Proxy, Cookie, Set-Cookie. It must be no more than 255 characters in length. Header name must be unique.

**21.1.8. `.spec.httpHeaders.actions.request[].action`****Description**

RouteHTTPHeaderActionUnion specifies an action to take on an HTTP header.

**Type****object****Required**

- **type**

Property	Type	Description
<b>set</b>	<b>object</b>	RouteSetHTTPHeader specifies what value needs to be set on an HTTP header.
<b>type</b>	<b>string</b>	type defines the type of the action to be applied on the header. Possible values are Set or Delete. Set allows you to set HTTP request and response headers. Delete allows you to delete HTTP request and response headers.

### 21.1.9. .spec.httpHeaders.actions.request[].action.set

#### Description

RouteSetHTTPHeader specifies what value needs to be set on an HTTP header.

#### Type

**object**

#### Required

- **value**

Property	Type	Description
<b>value</b>	<b>string</b>	value specifies a header value. Dynamic values can be added. The value will be interpreted as an HAProxy format string as defined in <a href="http://cbonte.github.io/haproxy-dconv/2.6/configuration.html#8.2.6">http://cbonte.github.io/haproxy-dconv/2.6/configuration.html#8.2.6</a> and may use HAProxy's %[] syntax and otherwise must be a valid HTTP header value as defined in <a href="https://datatracker.ietf.org/doc/html/rfc7230#section-3.2">https://datatracker.ietf.org/doc/html/rfc7230#section-3.2</a> . The value of this field must be no more than 16384 characters in length. Note that the total size of all net added headers <b>after</b> interpolating dynamic values must not exceed the value of spec.tuningOptions.headerBuffer MaxRewriteBytes on the IngressController.

## 21.1.10. .spec.httpHeaders.actions.response

### Description

response is a list of HTTP response headers to modify. Currently, actions may define to either **Set** or **Delete** headers values. Actions defined here will modify the response headers of all requests made through a route. These actions are applied to a specific Route defined within a cluster i.e. connections made through a route. Route actions will be executed before IngressController actions for response headers. Actions are applied in sequence as defined in this list. A maximum of 20 response header actions may be configured. You can use this field to specify HTTP response headers that should be set or deleted when forwarding responses from your application to the client. Sample fetchers allowed are "res.hdr" and "ssl\_c\_der". Converters allowed are "lower" and "base64". Example header values: "[%res.hdr(X-target),lower]", "%{+Q}[ssl\_c\_der,base64]". Note: This field cannot be used if your route uses TLS passthrough.

### Type

**array**

## 21.1.11. .spec.httpHeaders.actions.response[]

### Description

RouteHTTPHeader specifies configuration for setting or deleting an HTTP header.

### Type

**object**

### Required

- **name**
- **action**

Property	Type	Description
<b>action</b>	<b>object</b>	RouteHTTPHeaderActionUnion specifies an action to take on an HTTP header.
<b>name</b>	<b>string</b>	name specifies the name of a header on which to perform an action. Its value must be a valid HTTP header name as defined in RFC 2616 section 4.2. The name must consist only of alphanumeric and the following special characters, "-!#\$%&'*+.^_`". The following header names are reserved and may not be modified via this API: Strict-Transport-Security, Proxy, Cookie, Set-Cookie. It must be no more than 255 characters in length. Header name must be unique.



### 21.1.12. .spec.httpHeaders.actions.response[].action

#### Description

RouteHTTPHeaderActionUnion specifies an action to take on an HTTP header.

#### Type

**object**

#### Required

- **type**

Property	Type	Description
<b>set</b>	<b>object</b>	RouteSetHTTPHeader specifies what value needs to be set on an HTTP header.
<b>type</b>	<b>string</b>	type defines the type of the action to be applied on the header. Possible values are Set or Delete. Set allows you to set HTTP request and response headers. Delete allows you to delete HTTP request and response headers.

### 21.1.13. .spec.httpHeaders.actions.response[].action.set

#### Description

RouteSetHTTPHeader specifies what value needs to be set on an HTTP header.

#### Type

**object**

#### Required

- **value**

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

Property	Type	Description
<b>value</b>	<b>string</b>	<p>value specifies a header value. Dynamic values can be added. The value will be interpreted as an HAProxy format string as defined in <a href="http://cbonte.github.io/haproxy-dconv/2.6/configuration.html#8.2.6">http://cbonte.github.io/haproxy-dconv/2.6/configuration.html#8.2.6</a> and may use HAProxy's %[] syntax and otherwise must be a valid HTTP header value as defined in <a href="https://datatracker.ietf.org/doc/html/rfc7230#section-3.2">https://datatracker.ietf.org/doc/html/rfc7230#section-3.2</a>. The value of this field must be no more than 16384 characters in length. Note that the total size of all net added headers <b>after</b> interpolating dynamic values must not exceed the value of spec.tuningOptions.headerBufferMaxRewriteBytes on the IngressController.</p>

#### 21.1.14. .spec.port

##### Description

RoutePort defines a port mapping from a router to an endpoint in the service endpoints.

##### Type

**object**

##### Required

- **targetPort**

Property	Type	Description
<b>targetPort</b>	<b>IntOrString</b>	The target port on pods selected by the service this route points to. If this is a string, it will be looked up as a named port in the target endpoints port list. Required

#### 21.1.15. .spec.tls

##### Description

TLSConfig defines config used to secure a route and provide termination

##### Type

**object**

## Required

- **termination**

Property	Type	Description
<b>caCertificate</b>	<b>string</b>	caCertificate provides the cert authority certificate contents
<b>certificate</b>	<b>string</b>	certificate provides certificate contents. This should be a single serving certificate, not a certificate chain. Do not include a CA certificate.
<b>destinationCACertificate</b>	<b>string</b>	destinationCACertificate provides the contents of the ca certificate of the final destination. When using reencrypt termination this file should be provided in order to have routers use it for health checks on the secure connection. If this field is not specified, the router may provide its own destination CA and perform hostname validation using the short service name (service.namespace.svc), which allows infrastructure generated certificates to automatically verify.
<b>externalCertificate</b>	<b>object</b>	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Property	Type	Description
<b>insecureEdgeTerminationPolicy</b>	<b>string</b>	<p>insecureEdgeTerminationPolicy indicates the desired behavior for insecure connections to a route. While each router may make its own decisions on which ports to expose, this is normally port 80.</p> <p>If a route does not specify insecureEdgeTerminationPolicy, then the default behavior is "None".</p> <p>* Allow - traffic is sent to the server on the insecure port (edge/reencrypt terminations only).</p> <p>* None - no traffic is allowed on the insecure port (default).</p> <p>* Redirect - clients are redirected to the secure port.</p>
<b>key</b>	<b>string</b>	key provides key file contents
<b>termination</b>	<b>string</b>	<p>termination indicates termination type.</p> <p>* edge - TLS termination is done by the router and http is used to communicate with the backend (default) * passthrough - Traffic is sent straight to the destination without the router providing TLS termination * reencrypt - TLS termination is done by the router and https is used to communicate with the backend</p> <p>Note: passthrough termination is incompatible with httpHeader actions</p>

### 21.1.16. .spec.tls.externalCertificate

#### Description

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

#### Type

**object**

Property	Type	Description
<b>name</b>	<b>string</b>	name of the referent. More info: <a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a>

### 21.1.17. .spec.to

#### Description

RouteTargetReference specifies the target that resolve into endpoints. Only the 'Service' kind is allowed. Use 'weight' field to emphasize one over others.

#### Type

**object**

#### Required

- **kind**
- **name**

Property	Type	Description
<b>kind</b>	<b>string</b>	The kind of target that the route is referring to. Currently, only 'Service' is allowed
<b>name</b>	<b>string</b>	name of the service/target that is being referred to. e.g. name of the service
<b>weight</b>	<b>integer</b>	weight as an integer between 0 and 256, default 100, that specifies the target's relative weight against other target reference objects. 0 suppresses requests to this backend.

### 21.1.18. .status

#### Description

RouteStatus provides relevant info about the status of a route, including which routers acknowledge it.

#### Type

**object**

Property	Type	Description
<b>ingress</b>	<b>array</b>	ingress describes the places where the route may be exposed. The list of ingress points may contain duplicate Host or RouterName values. Routes are considered live once they are <b>Ready</b>
<b>ingress[]</b>	<b>object</b>	RouteIngress holds information about the places where a route is exposed.

### 21.1.19. .status.ingress

#### Description

ingress describes the places where the route may be exposed. The list of ingress points may contain duplicate Host or RouterName values. Routes are considered live once they are **Ready**

#### Type

**array**

### 21.1.20. .status.ingress[]

#### Description

RouteIngress holds information about the places where a route is exposed.

#### Type

**object**

Property	Type	Description
<b>conditions</b>	<b>array</b>	Conditions is the state of the route, may be empty.
<b>conditions[]</b>	<b>object</b>	RouteIngressCondition contains details for the current condition of this route on a particular router.
<b>host</b>	<b>string</b>	Host is the host string under which the route is exposed; this value is required
<b>routerCanonicalHostname</b>	<b>string</b>	CanonicalHostname is the external host name for the router that can be used as a CNAME for the host requested for this route. This value is optional and may not be set in all cases.

Property	Type	Description
<b>routerName</b>	<b>string</b>	Name is a name chosen by the router to identify itself; this value is required
<b>wildcardPolicy</b>	<b>string</b>	Wildcard policy is the wildcard policy that was allowed where this route is exposed.

### 21.1.21. .status.ingress[].conditions

#### Description

Conditions is the state of the route, may be empty.

#### Type

**array**

### 21.1.22. .status.ingress[].conditions[]

#### Description

RouteIngressCondition contains details for the current condition of this route on a particular router.

#### Type

**object**

#### Required

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

Property	Type	Description
<b>lastTransitionTime</b>	<b>Time</b>	RFC 3339 date and time when this condition last transitioned
<b>message</b>	<b>string</b>	Human readable message indicating details about last transition.
<b>reason</b>	<b>string</b>	(brief) reason for the condition's last transition, and is usually a machine and human readable constant
<b>status</b>	<b>string</b>	Status is the status of the condition. Can be True, False, Unknown.

Property	Type	Description
<b>type</b>	<b>string</b>	Type is the type of the condition. Currently only Admitted or UnservedInFutureVersions.

## 21.2. API ENDPOINTS

The following API endpoints are available:

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



### 21.2.1. /apis/route.openshift.io/v1/routes

HTTP method

**GET**

Description

list or watch objects of kind Route

Table 21.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">RouteList</a> schema
401 - Unauthorized	Empty

### 21.2.2. /apis/route.openshift.io/v1/watch/routes

HTTP method

**GET**

Description

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

Table 21.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

### 21.2.3. /apis/route.openshift.io/v1/namespaces/{namespace}/routes

HTTP method

**DELETE**

Description

delete collection of Route

Table 21.3. Query parameters

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

Table 21.4. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status_v7</a> schema
401 - Unauthorized	Empty

## HTTP method

**GET**

## Description

list or watch objects of kind Route

Table 21.5. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">RouteList</a> schema
401 - Unauthorized	Empty

## HTTP method

**POST**

## Description

create a Route

Table 21.6. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 21.7. Body parameters

Parameter	Type	Description
<b>body</b>	<b>Route</b> schema	

Table 21.8. HTTP responses

HTTP code	Response body
200 - OK	<b>Route</b> schema
201 - Created	<b>Route</b> schema
202 - Accepted	<b>Route</b> schema
401 - Unauthorized	Empty

#### 21.2.4. /apis/route.openshift.io/v1/watch/namespaces/{namespace}/routes

##### HTTP method

##### GET

##### Description

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

Table 21.9. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

### 21.2.5. /apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}

Table 21.10. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Route

#### HTTP method

#### DELETE

#### Description

delete a Route

Table 21.11. Query parameters

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

Table 21.12. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status_v7</a> schema
202 - Accepted	<a href="#">Status_v7</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

read the specified Route

Table 21.13. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Route</a> schema
401 - Unauthorized	Empty

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

partially update the specified Route

**Table 21.14. Query parameters**

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

**Table 21.15. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Route</a> schema
201 - Created	<a href="#">Route</a> schema

HTTP code	Response body
401 - Unauthorized	Empty

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

replace the specified Route

**Table 21.16. Query parameters**

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

**Table 21.17. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>Route</b> schema	

**Table 21.18. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Route</a> schema
201 - Created	<a href="#">Route</a> schema
401 - Unauthorized	Empty

### 21.2.6. /apis/route.openshift.io/v1/watch/namespaces/{namespace}/routes/{name}

Table 21.19. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Route

#### HTTP method

#### GET

#### Description

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

Table 21.20. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

### 21.2.7. /apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}/status

Table 21.21. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Route

#### HTTP method

#### GET

#### Description

read status of the specified Route

Table 21.22. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">Route</a> schema
401 - Unauthorized	Empty

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

partially update status of the specified Route

**Table 21.23. Query parameters**

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

**Table 21.24. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">Route</a> schema
201 - Created	<a href="#">Route</a> schema



HTTP code	Response body
401 - Unauthorized	Empty

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

replace status of the specified Route

**Table 21.25. Query parameters**

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

**Table 21.26. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>Route</b> schema	

**Table 21.27. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Route</a> schema
201 - Created	<a href="#">Route</a> schema
401 - Unauthorized	Empty

## CHAPTER 22. SERVICE [V1]

### Description

Service is a named abstraction of software service (for example, mysql) consisting of local port (for example 3306) that the proxy listens on, and the selector that determines which pods will answer requests sent through the proxy.

### Type

**object**

## 22.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	ServiceSpec describes the attributes that a user creates on a service.
<b>status</b>	<b>object</b>	ServiceStatus represents the current status of a service.

## 22.1.1. .spec

### Description

ServiceSpec describes the attributes that a user creates on a service.

### Type

**object**

Property	Type	Description
<b>allocateLoadBalancerNodePorts</b>	<b>boolean</b>	allocateLoadBalancerNodePorts defines if NodePorts will be automatically allocated for services with type LoadBalancer. Default is "true". It may be set to "false" if the cluster load-balancer does not rely on NodePorts. If the caller requests specific NodePorts (by specifying a value), those requests will be respected, regardless of this field. This field may only be set for services with type LoadBalancer and will be cleared if the type is changed to any other type.

Property	Type	Description
<b>clusterIP</b>	<b>string</b>	<p>clusterIP is the IP address of the service and is usually assigned randomly. If an address is specified manually, is in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field may not be changed through updates unless the type field is also being changed to ExternalName (which requires this field to be blank) or the type field is being changed from ExternalName (in which case this field may optionally be specified, as describe above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName. More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies">https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies</a></p>

Property	Type	Description
<b>clusterIPs</b>	<b>array (string)</b>	<p>ClusterIPs is a list of IP addresses assigned to this service, and are usually assigned randomly. If an address is specified manually, is in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field may not be changed through updates unless the type field is also being changed to ExternalName (which requires this field to be empty) or the type field is being changed from ExternalName (in which case this field may optionally be specified, as describe above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName. If this field is not specified, it will be initialized from the clusterIP field. If this field is specified, clients must ensure that clusterIPs[0] and clusterIP have the same value.</p> <p>This field may hold a maximum of two entries (dual-stack IPs, in either order). These IPs must correspond to the values of the ipFamilies field. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field. More info:  <a href="https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies">https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies</a></p>

Property	Type	Description
<b>externalIPs</b>	<b>array (string)</b>	externalIPs is a list of IP addresses for which nodes in the cluster will also accept traffic for this service. These IPs are not managed by Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load-balancers that are not part of the Kubernetes system.
<b>externalName</b>	<b>string</b>	externalName is the external reference that discovery mechanisms will return as an alias for this service (e.g. a DNS CNAME record). No proxying will be involved. Must be a lowercase RFC-1123 hostname ( <a href="https://tools.ietf.org/html/rfc1123">https://tools.ietf.org/html/rfc1123</a> ) and requires <b>type</b> to be "ExternalName".

Property	Type	Description
<b>externalTrafficPolicy</b>	<b>string</b>	<p>externalTrafficPolicy describes how nodes distribute service traffic they receive on one of the Service's "externally-facing" addresses (NodePorts, ExternalIPs, and LoadBalancer IPs). If set to "Local", the proxy will configure the service in a way that assumes that external load balancers will take care of balancing the service traffic between nodes, and so each node will deliver traffic only to the node-local endpoints of the service, without masquerading the client source IP. (Traffic mistakenly sent to a node with no endpoints will be dropped.) The default value, "Cluster", uses the standard behavior of routing to all endpoints evenly (possibly modified by topology and other features). Note that traffic sent to an External IP or LoadBalancer IP from within the cluster will always get "Cluster" semantics, but clients sending to a NodePort from within the cluster may need to take traffic policy into account when picking a node.</p> <p>Possible enum values: -  <b>"Cluster"</b> routes traffic to all endpoints. - <b>"Local"</b> preserves the source IP of the traffic by routing only to endpoints on the same node as the traffic was received on (dropping the traffic if there are no local endpoints).</p>



Property	Type	Description
<b>healthCheckNodePort</b>	<b>integer</b>	<p>healthCheckNodePort specifies the healthcheck nodePort for the service. This only applies when type is set to LoadBalancer and externalTrafficPolicy is set to Local. If a value is specified, is in-range, and is not in use, it will be used. If not specified, a value will be automatically allocated. External systems (e.g. load-balancers) can use this port to determine if a given node holds endpoints for this service or not. If this field is specified when creating a Service which does not need it, creation will fail. This field will be wiped when updating a Service to no longer need it (e.g. changing type). This field cannot be updated once set.</p>
<b>internalTrafficPolicy</b>	<b>string</b>	<p>InternalTrafficPolicy describes how nodes distribute service traffic they receive on the ClusterIP. If set to "Local", the proxy will assume that pods only want to talk to endpoints of the service on the same node as the pod, dropping the traffic if there are no local endpoints. The default value, "Cluster", uses the standard behavior of routing to all endpoints evenly (possibly modified by topology and other features).</p> <p>Possible enum values: -  <b>"Cluster"</b> routes traffic to all endpoints. - <b>"Local"</b> routes traffic only to endpoints on the same node as the client pod (dropping the traffic if there are no local endpoints).</p>

Property	Type	Description
<b>ipFamilies</b>	<b>array (string)</b>	<p>IPFamilies is a list of IP families (e.g. IPv4, IPv6) assigned to this service. This field is usually assigned automatically based on cluster configuration and the ipFamilyPolicy field. If this field is specified manually, the requested family is available in the cluster, and ipFamilyPolicy allows it, it will be used; otherwise creation of the service will fail. This field is conditionally mutable: it allows for adding or removing a secondary IP family, but it does not allow changing the primary IP family of the Service. Valid values are "IPv4" and "IPv6". This field only applies to Services of types ClusterIP, NodePort, and LoadBalancer, and does apply to "headless" services. This field will be wiped when updating a Service to type ExternalName.</p> <p>This field may hold a maximum of two entries (dual-stack families, in either order). These families must correspond to the values of the clusterIPs field, if specified. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field.</p>
<b>ipFamilyPolicy</b>	<b>string</b>	<p>IPFamilyPolicy represents the dual-stack-ness requested or required by this Service. If there is no value provided, then this field will be set to SingleStack. Services can be "SingleStack" (a single IP family), "PreferDualStack" (two IP families on dual-stack configured clusters or a single IP family on single-stack clusters), or "RequireDualStack" (two IP families on dual-stack configured clusters, otherwise fail). The ipFamilies and clusterIPs fields depend on the value of this field. This field will be wiped when updating a service to type ExternalName.</p>

Property	Type	Possible enum values: - Description
		<p><b>"PreferDualStack"</b> indicates that this service prefers dual-stack when the cluster is configured for dual-stack. If the cluster is not configured for dual-stack the service will be assigned a single IPFamily. If the IPFamily is not set in service.spec.ipFamilies then the service will be assigned the default IPFamily configured on the cluster -</p> <p><b>"RequireDualStack"</b> indicates that this service requires dual-stack. Using IPFamilyPolicyRequireDualStack on a single stack cluster will result in validation errors. The IPFamilies (and their order) assigned to this service is based on service.spec.ipFamilies. If service.spec.ipFamilies was not provided then it will be assigned according to how they are configured on the cluster. If service.spec.ipFamilies has only one entry then the alternative IPFamily will be added by apiserver -</p> <p><b>"SingleStack"</b> indicates that this service is required to have a single IPFamily. The IPFamily assigned is based on the default IPFamily used by the cluster or as identified by service.spec.ipFamilies field</p>

Property	Type	Description
<b>loadBalancerClass</b>	<b>string</b>	<p>loadBalancerClass is the class of the load balancer implementation this Service belongs to. If specified, the value of this field must be a label-style identifier, with an optional prefix, e.g. "internal-vip" or "example.com/internal-vip". Unprefixed names are reserved for end-users. This field can only be set when the Service type is 'LoadBalancer'. If not set, the default load balancer implementation is used, today this is typically done through the cloud provider integration, but should apply for any default implementation. If set, it is assumed that a load balancer implementation is watching for Services with a matching class. Any default load balancer implementation (e.g. cloud providers) should ignore Services that set this field. This field can only be set when creating or updating a Service to type 'LoadBalancer'. Once set, it can not be changed. This field will be wiped when a service is updated to a non 'LoadBalancer' type.</p>
<b>loadBalancerIP</b>	<b>string</b>	<p>Only applies to Service Type: LoadBalancer. This feature depends on whether the underlying cloud-provider supports specifying the loadBalancerIP when a load balancer is created. This field will be ignored if the cloud-provider does not support the feature. Deprecated: This field was underspecified and its meaning varies across implementations. Using it is non-portable and it may not support dual-stack. Users are encouraged to use implementation-specific annotations when available.</p>

Property	Type	Description
<b>loadBalancerSourceRanges</b>	<b>array (string)</b>	<p>If specified and supported by the platform, this will restrict traffic through the cloud-provider load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature."</p> <p>More info:  <a href="https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/">https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/</a></p>
<b>ports</b>	<b>array</b>	<p>The list of ports that are exposed by this service. More info:  <a href="https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies">https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies</a></p>
<b>ports[]</b>	<b>object</b>	ServicePort contains information on service's port.
<b>publishNotReadyAddresses</b>	<b>boolean</b>	<p>publishNotReadyAddresses indicates that any agent which deals with endpoints for this Service should disregard any indications of ready/not-ready. The primary use case for setting this field is for a StatefulSet's Headless Service to propagate SRV DNS records for its Pods for the purpose of peer discovery. The Kubernetes controllers that generate Endpoints and EndpointSlice resources for Services interpret this to mean that all endpoints are considered "ready" even if the Pods themselves are not. Agents which consume only Kubernetes generated endpoints through the Endpoints or EndpointSlice resources can safely assume this behavior.</p>

Property	Type	Description
<b>selector</b>	<b>object (string)</b>	Route service traffic to pods with label keys and values matching this selector. If empty or not present, the service is assumed to have an external process managing its endpoints, which Kubernetes will not modify. Only applies to types ClusterIP, NodePort, and LoadBalancer. Ignored if type is ExternalName. More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/">https://kubernetes.io/docs/concepts/services-networking/service/</a>
<b>sessionAffinity</b>	<b>string</b>	Supports "ClientIP" and "None". Used to maintain session affinity. Enable client IP based session affinity. Must be ClientIP or None. Defaults to None. More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies">https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies</a>  Possible enum values: - - <b>"ClientIP"</b> is the Client IP based. - <b>"None"</b> - no session affinity.
<b>sessionAffinityConfig</b>	<b>object</b>	SessionAffinityConfig represents the configurations of session affinity.
<b>trafficDistribution</b>	<b>string</b>	TrafficDistribution offers a way to express preferences for how traffic is distributed to Service endpoints. Implementations can use this field as a hint, but are not required to guarantee strict adherence. If the field is not set, the implementation will apply its default routing strategy. If set to "PreferClose", implementations should prioritize endpoints that are topologically close (e.g., same zone). This is an alpha field and requires enabling ServiceTrafficDistribution feature.

Property	Type	Description
<b>type</b>	<b>string</b>	<p>type determines how the Service is exposed. Defaults to ClusterIP. Valid options are ExternalName, ClusterIP, NodePort, and LoadBalancer. "ClusterIP" allocates a cluster-internal IP address for load-balancing to endpoints. Endpoints are determined by the selector or if that is not specified, by manual construction of an Endpoints object or EndpointSlice objects. If clusterIP is "None", no virtual IP is allocated and the endpoints are published as a set of endpoints rather than a virtual IP.</p> <p>"NodePort" builds on ClusterIP and allocates a port on every node which routes to the same endpoints as the clusterIP.</p> <p>"LoadBalancer" builds on NodePort and creates an external load-balancer (if supported in the current cloud) which routes to the same endpoints as the clusterIP.</p> <p>"ExternalName" aliases this service to the specified externalName. Several other fields do not apply to ExternalName services. More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types">https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types</a></p> <p>Possible enum values: -</p> <p><b>"ClusterIP"</b> means a service will only be accessible inside the cluster, via the cluster IP. -</p> <p><b>"ExternalName"</b> means a service consists of only a reference to an external name that kubedns or equivalent will return as a CNAME record, with no exposing or proxying of any pods involved. -</p> <p><b>"LoadBalancer"</b> means a service will be exposed via an external load balancer (if the cloud provider supports it), in addition to 'NodePort' type. -</p> <p><b>"NodePort"</b> means a service will</p>

Property	Type	Description
		be exposed on one port of every node, in addition to 'ClusterIP' type.

### 22.1.2. .spec.ports

#### Description

The list of ports that are exposed by this service. More info:

<https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

#### Type

**array**

### 22.1.3. .spec.ports[]

#### Description

ServicePort contains information on service's port.

#### Type

**object**

#### Required

- **port**

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



Property	Type	Description
<b>appProtocol</b>	<b>string</b>	<p>The application protocol for this port. This is used as a hint for implementations to offer richer behavior for protocols that they understand. This field follows standard Kubernetes label syntax. Valid values are either:</p> <ul style="list-style-type: none"> <li>* Un-prefixed protocol names - reserved for IANA standard service names (as per RFC-6335 and <a href="https://www.iana.org/assignments/service-names/">https://www.iana.org/assignments/service-names/</a>).</li> <li>* Kubernetes-defined prefixed names: <ul style="list-style-type: none"> <li>* 'kubernetes.io/h2c' - HTTP/2 prior knowledge over cleartext as described in <a href="https://www.rfc-editor.org/rfc/rfc9113.html#name-starting-http-2-with-prior-">https://www.rfc-editor.org/rfc/rfc9113.html#name-starting-http-2-with-prior-</a></li> <li>* 'kubernetes.io/ws' - WebSocket over cleartext as described in <a href="https://www.rfc-editor.org/rfc/rfc6455">https://www.rfc-editor.org/rfc/rfc6455</a></li> <li>* 'kubernetes.io/wss' - WebSocket over TLS as described in <a href="https://www.rfc-editor.org/rfc/rfc6455">https://www.rfc-editor.org/rfc/rfc6455</a></li> </ul> </li> <li>* Other protocols should use implementation-defined prefixed names such as mycompany.com/my-custom-protocol.</li> </ul>
<b>name</b>	<b>string</b>	<p>The name of this port within the service. This must be a DNS_LABEL. All ports within a ServiceSpec must have unique names. When considering the endpoints for a Service, this must match the 'name' field in the EndpointPort. Optional if only one ServicePort is defined on this service.</p>

Property	Type	Description
<b>nodePort</b>	<b>integer</b>	<p>The port on each node on which this service is exposed when type is NodePort or LoadBalancer. Usually assigned by the system. If a value is specified, in-range, and not in use it will be used, otherwise the operation will fail. If not specified, a port will be allocated if this Service requires one. If this field is specified when creating a Service which does not need it, creation will fail. This field will be wiped when updating a Service to no longer need it (e.g. changing type from NodePort to ClusterIP). More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport">https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport</a></p>
<b>port</b>	<b>integer</b>	The port that will be exposed by this service.
<b>protocol</b>	<b>string</b>	<p>The IP protocol for this port. Supports "TCP", "UDP", and "SCTP". Default is TCP.</p> <p>Possible enum values: - <b>"SCTP"</b> is the SCTP protocol. - <b>"TCP"</b> is the TCP protocol. - <b>"UDP"</b> is the UDP protocol.</p>

Property	Type	Description
<b>targetPort</b>	<b>IntOrString</b>	Number or name of the port to access on the pods targeted by the service. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME. If this is a string, it will be looked up as a named port in the target Pod's container ports. If this is not specified, the value of the 'port' field is used (an identity map). This field is ignored for services with clusterIP=None, and should be omitted or set equal to the 'port' field. More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service">https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service</a>

#### 22.1.4. .spec.sessionAffinityConfig

##### Description

SessionAffinityConfig represents the configurations of session affinity.

##### Type

**object**

Property	Type	Description
<b>clientIP</b>	<b>object</b>	ClientIPConfig represents the configurations of Client IP based session affinity.

#### 22.1.5. .spec.sessionAffinityConfig.clientIP

##### Description

ClientIPConfig represents the configurations of Client IP based session affinity.

##### Type

**object**

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

Property	Type	Description
<b>timeoutSeconds</b>	<b>integer</b>	timeoutSeconds specifies the seconds of ClientIP type session sticky time. The value must be >0 && ≤86400(for 1 day) if ServiceAffinity == "ClientIP". Default value is 10800(for 3 hours).

### 22.1.6. .status

#### Description

ServiceStatus represents the current status of a service.

#### Type

**object**

Property	Type	Description
<b>conditions</b>	<b>array (Condition)</b>	Current service state
<b>loadBalancer</b>	<b>object</b>	LoadBalancerStatus represents the status of a load-balancer.

### 22.1.7. .status.loadBalancer

#### Description

LoadBalancerStatus represents the status of a load-balancer.

#### Type

**object**

Property	Type	Description
<b>ingress</b>	<b>array</b>	Ingress is a list containing ingress points for the load-balancer. Traffic intended for the service should be sent to these ingress points.
<b>ingress[]</b>	<b>object</b>	LoadBalancerIngress represents the status of a load-balancer ingress point: traffic intended for the service should be sent to an ingress point.

### 22.1.8. .status.loadBalancer.ingress

**Description**

Ingress is a list containing ingress points for the load-balancer. Traffic intended for the service should be sent to these ingress points.

**Type**

**array**

**22.1.9. .status.loadBalancer.ingress[]****Description**

LoadBalancerIngress represents the status of a load-balancer ingress point: traffic intended for the service should be sent to an ingress point.

**Type**

**object**

Property	Type	Description
<b>hostname</b>	<b>string</b>	Hostname is set for load-balancer ingress points that are DNS based (typically AWS load-balancers)
<b>ip</b>	<b>string</b>	IP is set for load-balancer ingress points that are IP based (typically GCE or OpenStack load-balancers)
<b>ipMode</b>	<b>string</b>	IPMode specifies how the load-balancer IP behaves, and may only be specified when the ip field is specified. Setting this to "VIP" indicates that traffic is delivered to the node with the destination set to the load-balancer's IP and port. Setting this to "Proxy" indicates that traffic is delivered to the node or pod with the destination set to the node's IP and node port or the pod's IP and port. Service implementations may use this information to adjust traffic routing.
<b>ports</b>	<b>array</b>	Ports is a list of records of service ports. If used, every port defined in the service should have an entry in it.
<b>ports[]</b>	<b>object</b>	

**22.1.10. .status.loadBalancer.ingress[].ports**

**Description**

Ports is a list of records of service ports. If used, every port defined in the service should have an entry in it.

**Type**

**array**

**22.1.11. .status.loadBalancer.ingress[].ports[]****Description****Type**

**object**

**Required**

- **port**
- **protocol**

Property	Type	Description
<b>error</b>	<b>string</b>	Error is to record the problem with the service port. The format of the error shall comply with the following rules: - built-in error values shall be specified in this file and those shall use CamelCase names - cloud provider specific error values must have names that comply with the format foo.example.com/CamelCase.
<b>port</b>	<b>integer</b>	Port is the port number of the service port of which status is recorded here.
<b>protocol</b>	<b>string</b>	Protocol is the protocol of the service port of which status is recorded here. The supported values are: "TCP", "UDP", "SCTP".  Possible enum values: - <b>"SCTP"</b> is the SCTP protocol. - <b>"TCP"</b> is the TCP protocol. - <b>"UDP"</b> is the UDP protocol.

**22.2. API ENDPOINTS**

The following API endpoints are available:

- **/api/v1/services**
  - o **GET**: list or watch objects of kind Service

◦ **GET**: list or watch objects of kind Service

- **/api/v1/watch/services**

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

- **/api/v1/namespaces/{namespace}/services**

- **DELETE**: delete collection of Service
- **GET**: list or watch objects of kind Service
- **POST**: create a Service

- **/api/v1/watch/namespaces/{namespace}/services**

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

- **/api/v1/namespaces/{namespace}/services/{name}**

- **DELETE**: delete a Service
- **GET**: read the specified Service
- **PATCH**: partially update the specified Service
- **PUT**: replace the specified Service

- **/api/v1/watch/namespaces/{namespace}/services/{name}**

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

- **/api/v1/namespaces/{namespace}/services/{name}/status**

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

### 22.2.1. /api/v1/services

HTTP method

**GET**

Description

list or watch objects of kind Service

Table 22.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">ServiceList</a> schema

HTTP code	Reponse body
401 - Unauthorized	Empty

### 22.2.2. /api/v1/watch/services

#### HTTP method

#### GET

#### Description

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

Table 22.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

### 22.2.3. /api/v1/namespaces/{namespace}/services

#### HTTP method

#### DELETE

#### Description

delete collection of Service

Table 22.3. Query parameters

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

Table 22.4. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method



**GET****Description**

list or watch objects of kind Service

**Table 22.5. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">ServiceList</a> schema
401 - Unauthorized	Empty

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

create a Service

**Table 22.6. Query parameters**

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

**Table 22.7. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">Service</a> schema	

Table 22.8. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Service</a> schema
201 - Created	<a href="#">Service</a> schema
202 - Accepted	<a href="#">Service</a> schema
401 - Unauthorized	Empty

#### 22.2.4. /api/v1/watch/namespaces/{namespace}/services

##### HTTP method

##### GET

##### Description

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

Table 22.9. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

#### 22.2.5. /api/v1/namespaces/{namespace}/services/{name}

Table 22.10. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Service

##### HTTP method

##### DELETE

##### Description

delete a Service

Table 22.11. Query parameters

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

Table 22.12. HTTP responses

HTTP code	Response body
200 - OK	<b>Service</b> schema
202 - Accepted	<b>Service</b> schema
401 - Unauthorized	Empty

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

read the specified Service

Table 22.13. HTTP responses

HTTP code	Response body
200 - OK	<b>Service</b> schema
401 - Unauthorized	Empty

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

partially update the specified Service

Table 22.14. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive 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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 22.15. HTTP responses

HTTP code	Response body
200 - OK	<b>Service</b> schema
201 - Created	<b>Service</b> schema
401 - Unauthorized	Empty

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

replace the specified Service

Table 22.16. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+</li> <li>- 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.</li> </ul>

Table 22.17. Body parameters

Parameter	Type	Description
<b>body</b>	<b>Service</b> schema	

Table 22.18. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Service</b> schema
201 - Created	<b>Service</b> schema
401 - Unauthorized	Empty

### 22.2.6. /api/v1/watch/namespaces/{namespace}/services/{name}

Table 22.19. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Service

HTTP method

**GET****Description**

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

**Table 22.20. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">WatchEvent</a> schema
401 - Unauthorized	Empty

**22.2.7. /api/v1/namespaces/{namespace}/services/{name}/status****Table 22.21. Global path parameters**

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the Service

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

read status of the specified Service

**Table 22.22. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">Service</a> schema
401 - Unauthorized	Empty

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

partially update status of the specified Service

**Table 22.23. Query parameters**

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

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

Table 22.24. HTTP responses

HTTP code	Response body
200 - OK	<b>Service</b> schema
201 - Created	<b>Service</b> schema
401 - Unauthorized	Empty

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

replace status of the specified Service

Table 22.25. Query parameters

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

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

Table 22.26. Body parameters

Parameter	Type	Description
<b>body</b>	<b>Service</b> schema	

Table 22.27. HTTP responses

HTTP code	Response body
200 - OK	<b>Service</b> schema
201 - Created	<b>Service</b> schema
401 - Unauthorized	Empty