



# OpenShift Container Platform 4.17

## Cluster APIs

Reference guide for cluster APIs



# OpenShift Container Platform 4.17 Cluster APIs

---

Reference guide for cluster APIs

## Legal Notice

Copyright © 2024 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 cluster API objects and their detailed specifications.

## Table of Contents

<b>CHAPTER 1. CLUSTER APIS</b> .....	<b>3</b>
1.1. IPADDRESS [IPAM.CLUSTER.X-K8S.IO/V1BETA1]	3
1.2. IPADDRESSCLAIM [IPAM.CLUSTER.X-K8S.IO/V1BETA1]	3
<b>CHAPTER 2. IPADDRESS [IPAM.CLUSTER.X-K8S.IO/V1BETA1]</b> .....	<b>4</b>
2.1. SPECIFICATION	4
2.1.1. .spec	4
2.1.2. .spec.claimRef	5
2.1.3. .spec.poolRef	5
2.2. API ENDPOINTS	6
2.2.1. /apis/ipam.cluster.x-k8s.io/v1beta1/ipaddresses	6
2.2.2. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses	7
2.2.3. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses/{name}	8
<b>CHAPTER 3. IPADDRESSCLAIM [IPAM.CLUSTER.X-K8S.IO/V1BETA1]</b> .....	<b>12</b>
3.1. SPECIFICATION	12
3.1.1. .spec	13
3.1.2. .spec.poolRef	13
3.1.3. .status	13
3.1.4. .status.addressRef	14
3.1.5. .status.conditions	14
3.1.6. .status.conditions[]	14
3.2. API ENDPOINTS	15
3.2.1. /apis/ipam.cluster.x-k8s.io/v1beta1/ipaddressclaims	16
3.2.2. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims	16
3.2.3. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims/{name}	18
3.2.4. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims/{name}/status	21



## CHAPTER 1. CLUSTER APIS

### 1.1. IPADDRESS [IPAM.CLUSTER.X-K8S.IO/V1BETA1]

#### Description

IPAddress is the Schema for the ipaddress API.

#### Type

**object**

### 1.2. IPADDRESSCLAIM [IPAM.CLUSTER.X-K8S.IO/V1BETA1]

#### Description

IPAddressClaim is the Schema for the ipaddressclaim API.

#### Type

**object**

## CHAPTER 2. IPADDRESS [IPAM.CLUSTER.X-K8S.IO/V1BETA1]

### Description

IPAddress is the Schema for the ipaddress API.

### Type

**object**

## 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>	IPAddressSpec is the desired state of an IPAddress.

### 2.1.1. .spec

#### Description

IPAddressSpec is the desired state of an IPAddress.

#### Type

**object**



**Required**

- **address**
- **claimRef**
- **poolRef**
- **prefix**

Property	Type	Description
<b>address</b>	<b>string</b>	Address is the IP address.
<b>claimRef</b>	<b>object</b>	ClaimRef is a reference to the claim this IPAddress was created for.
<b>gateway</b>	<b>string</b>	Gateway is the network gateway of the network the address is from.
<b>poolRef</b>	<b>object</b>	PoolRef is a reference to the pool that this IPAddress was created from.
<b>prefix</b>	<b>integer</b>	Prefix is the prefix of the address.

**2.1.2. .spec.claimRef****Description**

ClaimRef is a reference to the claim this IPAddress was created for.

**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> TODO: Add other useful fields. apiVersion, kind, uid?

**2.1.3. .spec.poolRef****Description**

PoolRef is a reference to the pool that this IPAddress was created from.

**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
<b>name</b>	<b>string</b>	Name is the name of resource being referenced

## 2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/ipam.cluster.x-k8s.io/v1beta1/ipaddresses**
  - **GET**: list objects of kind IPAddress
- **/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses**
  - **DELETE**: delete collection of IPAddress
  - **GET**: list objects of kind IPAddress
  - **POST**: create an IPAddress
- **/apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses/{name}**
  - **DELETE**: delete an IPAddress
  - **GET**: read the specified IPAddress
  - **PATCH**: partially update the specified IPAddress
  - **PUT**: replace the specified IPAddress

### 2.2.1. /apis/ipam.cluster.x-k8s.io/v1beta1/ipaddresses

HTTP method

**GET****Description**

list objects of kind IPAddress

**Table 2.1. HTTP responses**

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

**2.2.2. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses****HTTP method****DELETE****Description**

delete collection of IPAddress

**Table 2.2. HTTP responses**

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

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

list objects of kind IPAddress

**Table 2.3. HTTP responses**

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

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

create an IPAddress

**Table 2.4. 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 2.5. Body parameters

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

Table 2.6. HTTP responses

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

### 2.2.3. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddresses/{name}

Table 2.7. Global path parameters

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

## HTTP method

**DELETE**

## Description

delete an IPAddress

Table 2.8. 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 2.9. 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 IPAddress

Table 2.10. HTTP responses

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

## HTTP method

**PATCH**

**Description**

partially update the specified IPAddress

**Table 2.11. 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 2.12. HTTP responses**

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

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

replace the specified IPAddress

**Table 2.13. 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 2.14. Body parameters

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

Table 2.15. HTTP responses

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

## CHAPTER 3. IPADDRESSCLAIM [IPAM.CLUSTER.X-K8S.IO/V1BETA1]

### Description

IPAddressClaim is the Schema for the ipaddressclaim API.

### Type

**object**

### 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>
<b>spec</b>	<b>object</b>	IPAddressClaimSpec is the desired state of an IPAddressClaim.
<b>status</b>	<b>object</b>	IPAddressClaimStatus is the observed status of a IPAddressClaim.



### 3.1.1. .spec

#### Description

IPAddressClaimSpec is the desired state of an IPAddressClaim.

#### Type

**object**

#### Required

- **poolRef**

Property	Type	Description
<b>poolRef</b>	<b>object</b>	PoolRef is a reference to the pool from which an IP address should be created.

### 3.1.2. .spec.poolRef

#### Description

PoolRef is a reference to the pool from which an IP address should be created.

#### 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
<b>name</b>	<b>string</b>	Name is the name of resource being referenced

### 3.1.3. .status

#### Description

IPAddressClaimStatus is the observed status of a IPAddressClaim.

**Type****object**

Property	Type	Description
<b>addressRef</b>	<b>object</b>	AddressRef is a reference to the address that was created for this claim.
<b>conditions</b>	<b>array</b>	Conditions summarises the current state of the IPAddressClaim
<b>conditions[]</b>	<b>object</b>	Condition defines an observation of a Cluster API resource operational state.

**3.1.4. .status.addressRef****Description**

AddressRef is a reference to the address that was created for this claim.

**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> TODO: Add other useful fields. apiVersion, kind, uid?

**3.1.5. .status.conditions****Description**

Conditions summarises the current state of the IPAddressClaim

**Type****array****3.1.6. .status.conditions[]****Description**

Condition defines an observation of a Cluster API resource operational state.

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

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

Property	Type	Description
<b>lastTransitionTime</b>	<b>string</b>	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>	A human readable message indicating details about the transition. This field may be empty.
<b>reason</b>	<b>string</b>	The reason for the condition's last transition in CamelCase. The specific API may choose whether or not this field is considered a guaranteed API. This field may not be empty.
<b>severity</b>	<b>string</b>	Severity provides an explicit classification of Reason code, so the users or machines can immediately understand the current situation and act accordingly. The Severity field MUST be set only when Status=False.
<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.

## 3.2. API ENDPOINTS

The following API endpoints are available:

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

### 3.2.1. /apis/ipam.cluster.x-k8s.io/v1beta1/ipaddressclaims

HTTP method

**GET**

Description

list objects of kind IPAddressClaim

Table 3.1. HTTP responses

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

### 3.2.2. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims

HTTP method

**DELETE**

Description

delete collection of IPAddressClaim

**Table 3.2. HTTP responses**

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

HTTP method

**GET**

Description

list objects of kind IPAddressClaim

**Table 3.3. HTTP responses**

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

HTTP method

**POST**

Description

create an IPAddressClaim

**Table 3.4. 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 3.5. Body parameters

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

Table 3.6. HTTP responses

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

### 3.2.3. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims/{name}

Table 3.7. Global path parameters

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

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

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

delete an IPAddressClaim

**Table 3.8. 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 3.9. 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 IPAddressClaim

**Table 3.10. HTTP responses**

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

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

partially update the specified IPAddressClaim

Table 3.11. 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 3.12. HTTP responses

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

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

replace the specified [IPAddressClaim](#)

Table 3.13. 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 3.14. Body parameters

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

Table 3.15. HTTP responses

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

### 3.2.4. /apis/ipam.cluster.x-k8s.io/v1beta1/namespaces/{namespace}/ipaddressclaims/{name}/status

Table 3.16. Global path parameters

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

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

read status of the specified IPAddressClaim

**Table 3.17. HTTP responses**

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

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

partially update status of the specified IPAddressClaim

**Table 3.18. 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 3.19. HTTP responses

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

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

replace status of the specified IPAddressClaim

Table 3.20. 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 3.21. Body parameters

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

Table 3.22. HTTP responses

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