Networking API v2.0

Layer 2 Networking¶

Networks¶

Lists, shows details for, creates, updates, and deletes networks.

Address Scopes Extension¶

The `address-scope` extension adds the `ipv4_address_scope` and `ipv6_address_scope` attributes to networks. `ipv4_address_scope` is the ID of the IPv4 address scope that the network is associated with. `ipv6_address_scope` is the ID of the IPv6 address scope that the network is associated with.

Auto Allocated Topology¶

The `auto-allocated-topology` extension adds the `is_default` boolean attribute to networks. This value indicates the network should be used when auto allocating topologies.

DNS integration¶

The `dns-integration` extension adds the `dns_domain` attribute to networks. The `dns_domain` of a network in conjunction with the `dns_name` attribute of its ports will be published in an external DNS service when Neutron is configured to integrate with such a service.

External network¶

The `external-net` extension adds the `router:external` attribute to networks. This boolean attribute indicates the network has an external routing facility that's not managed by the networking service.

L2 adjacency extension¶

The `l2_adjacency` extension provides display of L2 Adjacency for `networks` by adding the read-only `l2_adjacency` attribute. This is a boolean value where `true` means that you can expect L2 connectivity throughout the Network and `false` means that there is no guarantee of L2 connectivity. This value is read-only and is derived from the current state of `segments` within the `network`.

MTU extensions¶

The `net-mtu` extension allows plug-ins to expose the MTU that is guaranteed to pass through the data path of the segments in the network. This extension introduces a read-only `mtu` attribute.

A newer `net-mtu-writable` extension enhances `net-mtu` in that now the `mtu` attribute is available for write (both when creating as well as updating networks).

Multiple provider extension¶

The `multi-provider` extension allows administrative users to define multiple physical bindings for a logical network.

To define multiple physical bindings for a network, include a `segments` list in the request body of network creation request. Each element in the `segments` list has the same structure as the provider network attributes. These attributes are `provider:network_type`, `provider:physical_network`, and `provider:segmentation_id`. The same validation rules are applied to each element in the `segments` list.

Note that you cannot use the provider extension and the multiple provider extension for a single logical network.

Network availability zone extension¶

The `network_availability_zone` extension provides support of availability zone for networks, exposing `availability_zone_hints` and `availability_zones` attributes.

Port security¶

The `port-security` extension adds the `port_security_enabled` boolean attribute to networks. At the network level, `port_security_enabled` defines the default value for new ports attached to the network; they will inherit the value of their network's `port_security_enabled` unless explicitly set on the port itself. While the default value for `port_security_enabled` is `true`, this can be changed by updating the respective network. Note that changing a value of `port_security_enabled` on a network, does not cascade the value to ports attached to the network.

Provider extended attributes¶

The `provider` extension allows administrative users to define a physical binding of a logical network. This extension provides three additional attributes: `provider:network_type`, `provider:physical_network` and `provider:segmentation_id`. The validation rules for these attributes vary across `provider:network_type`. For example, `vlan` and `flat` network types require `provider:physical_network` attribute, but `vxlan` network type does not.

Most Networking plug-ins (e.g. ML2 Plugin) and drivers do not support updating any provider related attributes. Check your plug-in whether it supports updating.

QoS extension¶

The QoS extension (`qos`) makes it possible to define QoS policies and associate these to the networks by introducing the `qos_policy_id` attribute. The policies should be created before they are associated to the networks.

Resource timestamps¶

The `standard-attr-timestamp` extension adds the `created_at` and `updated_at` attributes to all resources that have standard attributes.

Tag extension¶

The `standard-attr-tag` adds Tag support for resources with standard attributes by adding the `tags` attribute allowing consumers to associate tags with resources.

VLAN transparency extension¶

The `vlan-transparent` extension enables plug-ins that support VLAN transparency to deliver VLAN transparent trunk networks. This extension introduces a `vlan_transparent` attribute to control the VLAN transparency of the network. If the service does not support VLAN transparency and a user requests a VLAN transparent network, the plug-in refuses to create one and returns an appropriate error to the user.

Show network details GET /v2.0/networks/{network_id}

Shows details for a network.

Use the `fields` query parameter to control which fields are returned in the response body. For information, see Filtering and Column Selection.

Normal response codes: 200

Error response codes: 401, 404

Request¶

Name	In	Type	Description
network_id	path	string	The ID of the network.
fields (Optional)	query	string	The fields that you want the server to return. If no `fields` query parameter is specified, the networking API returns all attributes allowed by the policy settings. By using `fields` parameter, the API returns only the requested set of attributes. `fields` parameter can be specified multiple times. For example, if you specify `fields=id&fields=name` in the request URL, only `id` and `name` attributes will be returned.

Response Parameters¶

Name	In	Type	Description
network	body	object	A `network` object.
admin_state_up	body	boolean	The administrative state of the network, which is up (`true`) or down (`false`).
availability_zone_hints	body	array	The availability zone candidate for the network.
availability_zones	body	array	The availability zone for the network.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
dns_domain	body	string	A valid DNS domain.
id	body	string	The ID of the network.
ipv4_address_scope	body	string	The ID of the IPv4 address scope that the network is associated with.
ipv6_address_scope	body	string	The ID of the IPv6 address scope that the network is associated with.
l2_adjacency	body	boolean	Indicates whether L2 connectivity is available throughout the `network`.
mtu	body	integer	The maximum transmission unit (MTU) value to address fragmentation. Minimum value is 68 for IPv4, and 1280 for IPv6.
name	body	string	Human-readable name of the network.
port_security_enabled	body	boolean	The port security status of the network. Valid values are enabled (`true`) and disabled (`false`). This value is used as the default value of `port_security_enabled` field of a newly created port.
project_id	body	string	The ID of the project.
provider:network_type	body	string	The type of physical network that this network is mapped to. For example, `flat`, `vlan`, `vxlan`, or `gre`. Valid values depend on a networking back-end.
provider:physical_network	body	string	The physical network where this network/segment is implemented.
provider:segmentation_id	body	integer	The ID of the isolated segment on the physical network. The `network_type` attribute defines the segmentation model. For example, if the `network_type` value is vlan, this ID is a vlan identifier. If the `network_type` value is gre, this ID is a gre key.
qos_policy_id	body	string	The ID of the QoS policy associated with the network.
revision_number	body	integer	The revision number of the resource.
router:external	body	boolean	Indicates whether the network has an external routing facility that's not managed by the networking service.
segments	body	array	A list of provider `segment` objects.
shared	body	boolean	Indicates whether this network is shared across all tenants. By default, only administrative users can change this value.
status	body	string	The network status. Values are `ACTIVE`, `DOWN`, `BUILD` or `ERROR`.
subnets	body	array	The associated subnets.
tenant_id	body	string	The ID of the project.
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
vlan_transparent	body	boolean	Indicates the VLAN transparency mode of the network, which is VLAN transparent (`true`) or not VLAN transparent (`false`).
description	body	string	A human-readable description for the resource.
is_default	body	boolean	The network is default pool or not.
tags	body	array	The list of tags on the resource.

Response Example¶

```
{
    "network": {
        "admin_state_up": true,
        "availability_zone_hints": [],
        "availability_zones": [
            "nova"
        ],
        "created_at": "2016-03-08T20:19:41",
        "dns_domain": "my-domain.org.",
        "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
        "ipv4_address_scope": null,
        "ipv6_address_scope": null,
        "l2_adjacency": false,
        "mtu": 1500,
        "name": "private-network",
        "port_security_enabled": true,
        "project_id": "4fd44f30292945e481c7b8a0c8908869",
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "revision_number": 1,
        "router:external": false,
        "shared": true,
        "status": "ACTIVE",
        "subnets": [
            "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
        ],
        "tags": ["tag1,tag2"],
        "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
        "updated_at": "2016-03-08T20:19:41",
        "vlan_transparent": false,
        "description": "",
        "is_default": true
    }
}
```

Response Example (admin user; single segment mapping)¶

```
{
    "network": {
        "admin_state_up": true,
        "availability_zone_hints": [],
        "availability_zones": [
            "nova"
        ],
        "created_at": "2016-03-08T20:19:41",
        "dns_domain": "my-domain.org.",
        "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
        "ipv4_address_scope": null,
        "ipv6_address_scope": null,
        "l2_adjacency": false,
        "mtu": 1500,
        "name": "private-network",
        "port_security_enabled": true,
        "project_id": "4fd44f30292945e481c7b8a0c8908869",
        "provider:network_type": "local",
        "provider:physical_network": null,
        "provider:segmentation_id": null,
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "revision_number": 1,
        "router:external": false,
        "shared": true,
        "status": "ACTIVE",
        "subnets": [
            "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
        ],
        "tags": ["tag1,tag2"],
        "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
        "updated_at": "2016-03-08T20:19:41",
        "vlan_transparent": false,
        "description": "",
        "is_default": true
    }
}
```

Response Example (admin user; multiple segment mappings)¶

```
{
    "network": {
        "admin_state_up": true,
        "availability_zone_hints": [],
        "availability_zones": [
            "nova"
        ],
        "created_at": "2016-03-08T20:19:41",
        "dns_domain": "my-domain.org.",
        "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c",
        "ipv4_address_scope": null,
        "ipv6_address_scope": null,
        "l2_adjacency": false,
        "mtu": 1500,
        "name": "net1",
        "port_security_enabled": true,
        "project_id": "9bacb3c5d39d41a79512987f338cf177",
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "revision_number": 1,
        "router:external": false,
        "segments": [
            {
                "provider:network_type": "vlan",
                "provider:physical_network": "public",
                "provider:segmentation_id": 2
            },
            {
                "provider:network_type": "flat",
                "provider:physical_network": "default",
                "provider:segmentation_id": 0
            }
        ],
        "shared": false,
        "status": "ACTIVE",
        "subnets": [
            "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
        ],
        "tags": ["tag1,tag2"],
        "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
        "updated_at": "2016-03-08T20:19:41",
        "vlan_transparent": false,
        "description": "",
        "is_default": false
    }
}
```

Update network PUT /v2.0/networks/{network_id}

Updates a network.

Normal response codes: 200

Error response codes: 400, 401, 403, 404, 412

Request¶

Name	In	Type	Description
network_id	path	string	The ID of the network.
network	body	object	A `network` object.
admin_state_up (Optional)	body	boolean	The administrative state of the network, which is up (`true`) or down (`false`).
dns_domain (Optional)	body	string	A valid DNS domain.
mtu (Optional)	body	integer	The maximum transmission unit (MTU) value to address fragmentation. Minimum value is 68 for IPv4, and 1280 for IPv6.
name (Optional)	body	string	Human-readable name of the network.
port_security_enabled (Optional)	body	boolean	The port security status of the network. Valid values are enabled (`true`) and disabled (`false`). This value is used as the default value of `port_security_enabled` field of a newly created port.
provider:network_type	body	string	The type of physical network that this network is mapped to. For example, `flat` , `vlan` , `vxlan` , or `gre` . Valid values depend on a networking back-end.
provider:physical_network	body	string	The physical network where this network/segment is implemented.
provider:segmentation_id	body	integer	The ID of the isolated segment on the physical network. The `network_type` attribute defines the segmentation model. For example, if the `network_type` value is vlan, this ID is a vlan identifier. If the `network_type` value is gre, this ID is a gre key.
qos_policy_id (Optional)	body	string	The ID of the QoS policy associated with the network.
router:external (Optional)	body	boolean	Indicates whether the network has an external routing facility that's not managed by the networking service.
segments	body	array	A list of provider `segment` objects.
shared (Optional)	body	boolean	Indicates whether this resource is shared across all projects. By default, only administrative users can change this value.
description (Optional)	body	string	A human-readable description for the resource. Default is an empty string.
is_default (Optional)	body	boolean	The network is default or not.

Request Example¶

```
{
    "network": {
        "dns_domain": "my-domain.org.",
        "name": "sample_network_5_updated",
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "mtu": 1300
    }
}
```

Request Example (admin user; single segment mapping)¶

```
{
    "network": {
        "provider:network_type": "vlan",
        "provider:physical_network": "public",
        "provider:segmentation_id": 2
    }
}
```

Request Example (admin user; multiple segment mappings)¶

```
{
    "network": {
        "segments": [
            {
                "provider:segmentation_id": 2,
                "provider:physical_network": "public",
                "provider:network_type": "vlan"
            },
            {
                "provider:physical_network": "default",
                "provider:network_type": "flat"
            }
        ]
    }
}
```

Response Parameters¶

Name	In	Type	Description
network	body	object	A `network` object.
admin_state_up	body	boolean	The administrative state of the network, which is up (`true`) or down (`false`).
availability_zone_hints	body	array	The availability zone candidate for the network.
availability_zones	body	array	The availability zone for the network.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
dns_domain	body	string	A valid DNS domain.
id	body	string	The ID of the network.
ipv4_address_scope	body	string	The ID of the IPv4 address scope that the network is associated with.
ipv6_address_scope	body	string	The ID of the IPv6 address scope that the network is associated with.
l2_adjacency	body	boolean	Indicates whether L2 connectivity is available throughout the `network` .
mtu	body	integer	The maximum transmission unit (MTU) value to address fragmentation. Minimum value is 68 for IPv4, and 1280 for IPv6.
name	body	string	Human-readable name of the network.
port_security_enabled	body	boolean	The port security status of the network. Valid values are enabled (`true`) and disabled (`false`). This value is used as the default value of `port_security_enabled` field of a newly created port.
project_id	body	string	The ID of the project.
provider:network_type	body	string	The type of physical network that this network is mapped to. For example, `flat` , `vlan` , `vxlan` , or `gre` . Valid values depend on a networking back-end.
provider:physical_network	body	string	The physical network where this network/segment is implemented.
provider:segmentation_id	body	integer	The ID of the isolated segment on the physical network. The `network_type` attribute defines the segmentation model. For example, if the `network_type` value is vlan, this ID is a vlan identifier. If the `network_type` value is gre, this ID is a gre key.
qos_policy_id	body	string	The ID of the QoS policy associated with the network.
revision_number	body	integer	The revision number of the resource.
router:external	body	boolean	Indicates whether the network has an external routing facility that's not managed by the networking service.
segments	body	array	A list of provider `segment` objects.
shared	body	boolean	Indicates whether this network is shared across all tenants. By default, only administrative users can change this value.
status	body	string	The network status. Values are `ACTIVE` , `DOWN` , `BUILD` or `ERROR` .
subnets	body	array	The associated subnets.
tenant_id	body	string	The ID of the project.
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
description	body	string	A human-readable description for the resource.
is_default	body	boolean	The network is default pool or not.
tags	body	array	The list of tags on the resource.

Response Example¶

This is an example when a regular user without administrative roles sends a PUT request. Response examples for administrative users are similar to responses of Show network details and Create network. See them for details.

```
{
    "network": {
        "admin_state_up": true,
        "availability_zone_hints": [],
        "availability_zones": [
            "nova"
        ],
        "created_at": "2016-03-08T20:19:41",
        "dns_domain": "my-domain.org.",
        "id": "1f370095-98f6-4079-be64-6d3d4a6adcc6",
        "ipv4_address_scope": null,
        "ipv6_address_scope": null,
        "l2_adjacency": false,
        "mtu": 1300,
        "name": "sample_network_5_updated",
        "port_security_enabled": true,
        "project_id": "4fd44f30292945e481c7b8a0c8908869",
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "revision_number": 2,
        "router:external": false,
        "shared": false,
        "status": "ACTIVE",
        "subnets": [
            "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
        ],
        "tags": ["tag1,tag2"],
        "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
        "updated_at": "2016-03-08T20:19:41",
        "vlan_transparent": false,
        "description": "",
        "is_default": false
    }
}
```

Delete network DELETE /v2.0/networks/{network_id}

Deletes a network and its associated resources.

Normal response codes: 204

Error response codes: 401, 404, 409, 412

Request¶

Name	In	Type	Description
network_id	path	string	The ID of the network.

Response¶

There is no body content for the response of a successful DELETE request.

List networks GET /v2.0/networks

Lists networks to which the project has access.

Default policy settings return only networks that the project who submits the request owns, unless an administrative user submits the request. In addition, networks shared with the project who submits the request are also returned.

Use the `fields` query parameter to control which fields are returned in the response body. Additionally, you can filter results by using query string parameters. For information, see Filtering and Column Selection.

You can also use the `tags`, `tags-any`, `not-tags`, `not-tags-any` query parameter to filter the response with tags. For information, see REST API Impact.

Normal response codes: 200

Error response codes: 401

Request¶

Name	In	Type	Description
admin_state_up (Optional)	query	boolean	Filter the list result by the administrative state of the resource, which is up (`true`) or down (`false`).
id (Optional)	query	string	Filter the list result by the ID of the resource.
mtu (Optional)	query	integer	Filter the network list result by the maximum transmission unit (MTU) value to address fragmentation. Minimum value is `68` for IPv4, and `1280` for IPv6.
name (Optional)	query	string	Filter the list result by the human-readable name of the resource.
project_id (Optional)	query	string	Filter the list result by the ID of the project that owns the resource.
provider:network_type (Optional)	query	string	Filter the list result by the type of physical network that this network/segment is mapped to. For example, `flat`, `vlan`, `vxlan`, or `gre`. Valid values depend on a networking back-end.
provider:physical_network (Optional)	query	string	Filter the list result by the physical network where this network/segment is implemented.
provider:segmentation_id (Optional)	query	integer	Filter the list result by the ID of the isolated segment on the physical network.
revision_number (Optional)	query	integer	Filter the list result by the revision number of the resource.
router:external (Optional)	query	boolean	Filter the network list result based on whether the network has an external routing facility that's not managed by the networking service.
shared (Optional)	query	boolean	Filter the network list result based on if the network is shared across all tenants.
status (Optional)	query	string	Filter the network list result by network status. Values are `ACTIVE`, `DOWN`, `BUILD` or `ERROR`.
tenant_id (Optional)	query	string	Filter the list result by the ID of the project that owns the resource.
vlan_transparent (Optional)	query	boolean	Filter the network list by the VLAN transparency mode of the network, which is VLAN transparent (`true`) or not VLAN transparent (`false`).
description (Optional)	query	string	Filter the list result by the human-readable description of the resource.
is_default (Optional)	query	boolean	Filter the network list result based on if the network is default pool or not.
tags (Optional)	query	string	A list of tags to filter the list result by. Resources that match all tags in this list will be returned. Tags in query must be separated by comma.
tags-any (Optional)	query	string	A list of tags to filter the list result by. Resources that match any tag in this list will be returned. Tags in query must be separated by comma.
not-tags (Optional)	query	string	A list of tags to filter the list result by. Resources that match all tags in this list will be excluded. Tags in query must be separated by comma.
not-tags-any (Optional)	query	string	A list of tags to filter the list result by. Resources that match any tag in this list will be excluded. Tags in query must be separated by comma.
sort_dir (Optional)	query	string	Sort direction. A valid value is `asc` (ascending) or `desc` (descending). You can specify multiple pairs of sort key and sort direction query parameters.
sort_key (Optional)	query	string	Sorts by a network attribute. You can specify multiple pairs of sort key and sort direction query parameters. The sort keys are limited to:```admin_state_up``availability_zone_hints``id``mtu``name``status``tenant_id``project_id```
fields (Optional)	query	string	The fields that you want the server to return. If no `fields` query parameter is specified, the networking API returns all attributes allowed by the policy settings. By using `fields` parameter, the API returns only the requested set of attributes. `fields` parameter can be specified multiple times. For example, if you specify `fields=id&fields=name` in the request URL, only `id` and `name` attributes will be returned.

Response Parameters¶

Name	In	Type	Description
networks	body	array	A list of `network` objects.
admin_state_up	body	boolean	The administrative state of the network, which is up (`true`) or down (`false`).
availability_zone_hints	body	array	The availability zone candidate for the network.
availability_zones	body	array	The availability zone for the network.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
dns_domain	body	string	A valid DNS domain.
id	body	string	The ID of the network.
ipv4_address_scope	body	string	The ID of the IPv4 address scope that the network is associated with.
ipv6_address_scope	body	string	The ID of the IPv6 address scope that the network is associated with.
l2_adjacency	body	boolean	Indicates whether L2 connectivity is available throughout the `network`.
mtu	body	integer	The maximum transmission unit (MTU) value to address fragmentation. Minimum value is 68 for IPv4, and 1280 for IPv6.
name	body	string	Human-readable name of the network.
port_security_enabled	body	boolean	The port security status of the network. Valid values are enabled (`true`) and disabled (`false`). This value is used as the default value of `port_security_enabled` field of a newly created port.
project_id	body	string	The ID of the project.
provider:network_type	body	string	The type of physical network that this network is mapped to. For example, `flat`, `vlan`, `vxlan`, or `gre`. Valid values depend on a networking back-end.
provider:physical_network	body	string	The physical network where this network/segment is implemented.
provider:segmentation_id	body	integer	The ID of the isolated segment on the physical network. The `network_type` attribute defines the segmentation model. For example, if the `network_type` value is vlan, this ID is a vlan identifier. If the `network_type` value is gre, this ID is a gre key.
qos_policy_id	body	string	The ID of the QoS policy associated with the network.
revision_number	body	integer	The revision number of the resource.
router:external	body	boolean	Indicates whether the network has an external routing facility that's not managed by the networking service.
segments	body	array	A list of provider `segment` objects.
shared	body	boolean	Indicates whether this network is shared across all tenants. By default, only administrative users can change this value.
status	body	string	The network status. Values are `ACTIVE`, `DOWN`, `BUILD` or `ERROR`.
subnets	body	array	The associated subnets.
tenant_id	body	string	The ID of the project.
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
vlan_transparent	body	boolean	Indicates the VLAN transparency mode of the network, which is VLAN transparent (`true`) or not VLAN transparent (`false`).
description	body	string	A human-readable description for the resource.
is_default	body	boolean	The network is default pool or not.
tags	body	array	The list of tags on the resource.

Response Example¶

```
{
    "networks": [
        {
            "admin_state_up": true,
            "availability_zone_hints": [],
            "availability_zones": [
                "nova"
            ],
            "created_at": "2016-03-08T20:19:41",
            "dns_domain": "my-domain.org.",
            "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
            "ipv4_address_scope": null,
            "ipv6_address_scope": null,
            "l2_adjacency": false,
            "mtu": 1500,
            "name": "net1",
            "port_security_enabled": true,
            "project_id": "4fd44f30292945e481c7b8a0c8908869",
            "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
            "revision_number": 1,
            "router:external": false,
            "shared": false,
            "status": "ACTIVE",
            "subnets": [
                "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
            ],
            "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
            "updated_at": "2016-03-08T20:19:41",
            "vlan_transparent": true,
            "description": "",
            "is_default": false
        },
        {
            "admin_state_up": true,
            "availability_zone_hints": [],
            "availability_zones": [
                "nova"
            ],
            "created_at": "2016-03-08T20:19:41",
            "dns_domain": "my-domain.org.",
            "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
            "ipv4_address_scope": null,
            "ipv6_address_scope": null,
            "l2_adjacency": false,
            "mtu": 1500,
            "name": "net2",
            "port_security_enabled": true,
            "project_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
            "qos_policy_id": "bfdb6c39f71e4d44b1dfbda245c50819",
            "revision_number": 3,
            "router:external": false,
            "shared": false,
            "status": "ACTIVE",
            "subnets": [
                "08eae331-0402-425a-923c-34f7cfe39c1b"
            ],
            "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
            "updated_at": "2016-03-08T20:19:41",
            "vlan_transparent": false,
            "description": "",
            "is_default": false
        }
    ]
}
```

Response Example (admin user)

When Administrative users request to list networks, physical segment information bound to the networks are also returned in a response. In this example, a network `net1` is mapped to a single network segment and a network `net2` is mapped to multiple network segments.

```
{
    "networks": [
        {
            "admin_state_up": true,
            "availability_zone_hints": [],
            "availability_zones": [
                "nova"
            ],
            "created_at": "2016-03-08T20:19:41",
            "dns_domain": "my-domain.org.",
            "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
            "ipv4_address_scope": null,
            "ipv6_address_scope": null,
            "l2_adjacency": false,
            "mtu": 1500,
            "name": "net1",
            "port_security_enabled": true,
            "project_id": "4fd44f30292945e481c7b8a0c8908869",
            "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
            "provider:network_type": "vlan",
            "provider:physical_network": "public",
            "provider:segmentation_id": 3,
            "revision_number": 1,
            "router:external": false,
            "shared": false,
            "status": "ACTIVE",
            "subnets": [
                "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
            ],
            "tags": ["tag1,tag2"],
            "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
            "updated_at": "2016-03-08T20:19:41",
            "vlan_transparent": true,
            "description": "",
            "is_default": false
        },
        {
            "admin_state_up": true,
            "availability_zone_hints": [],
            "availability_zones": [
                "nova"
            ],
            "created_at": "2016-03-08T20:19:41",
            "dns_domain": "my-domain.org.",
            "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
            "ipv4_address_scope": null,
            "ipv6_address_scope": null,
            "l2_adjacency": false,
            "mtu": 1450,
            "name": "net2",
            "port_security_enabled": true,
            "project_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
            "qos_policy_id": null,
            "provider:network_type": "local",
            "provider:physical_network": null,
            "provider:segmentation_id": null,
            "qos_policy_id": "bfdb6c39f71e4d44b1dfbda245c50819",
            "revision_number": 2,
            "router:external": false,
            "segments": [
                {
                    "provider:network_type": "vlan",
                    "provider:physical_network": "public",
                    "provider:segmentation_id": 2
                },
                {
                    "provider:network_type": "vxlan",
                    "provider:physical_network": "default",
                    "provider:segmentation_id": 1000
                }
            ],
            "shared": false,
            "status": "ACTIVE",
            "subnets": [
                "08eae331-0402-425a-923c-34f7cfe39c1b"
            ],
            "tags": ["tag1,tag2"],
            "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
            "updated_at": "2016-03-08T20:19:41",
            "vlan_transparent": false,
            "description": "",
            "is_default": false
        }
    ]
}
```

Create network POST /v2.0/networks

Creates a network.

A request body is optional. An administrative user can specify another project ID, which is the project that owns the network, in the request body.

Normal response codes: 201

Error response codes: 400, 401

Request¶

Name	In	Type	Description
network	body	object	A `network` object.
admin_state_up (Optional)	body	boolean	The administrative state of the network, which is up (`true`) or down (`false`).
dns_domain (Optional)	body	string	A valid DNS domain.
mtu (Optional)	body	integer	The maximum transmission unit (MTU) value to address fragmentation. Minimum value is 68 for IPv4, and 1280 for IPv6.
name (Optional)	body	string	Human-readable name of the network.
port_security_enabled (Optional)	body	boolean	The port security status of the network. Valid values are enabled (`true`) and disabled (`false`). This value is used as the default value of `port_security_enabled` field of a newly created port.
project_id (Optional)	body	string	The ID of the project that owns the resource. Only administrative and users with advsvc role can specify a project ID other than their own. You cannot change this value through authorization policies.
provider:network_type (Optional)	body	string	The type of physical network that this network should be mapped to. For example, `flat` , `vlan` , `vxlan` , or `gre` . Valid values depend on a networking back-end.
provider:physical_network (Optional)	body	string	The physical network where this network should be implemented. The Networking API v2.0 does not provide a way to list available physical networks. For example, the Open vSwitch plug-in configuration file defines a symbolic name that maps to specific bridges on each compute host.
provider:segmentation_id (Optional)	body	integer	The ID of the isolated segment on the physical network. The `network_type` attribute defines the segmentation model. For example, if the `network_type` value is vlan, this ID is a vlan identifier. If the `network_type` value is gre, this ID is a gre key.
qos_policy_id (Optional)	body	string	The ID of the QoS policy associated with the network.
router:external (Optional)	body	boolean	Indicates whether the network has an external routing facility that's not managed by the networking service.
segments (Optional)	body	array	A list of provider `segment` objects.
shared (Optional)	body	boolean	Indicates whether this resource is shared across all projects. By default, only administrative users can change this value.
tenant_id (Optional)	body	string	The ID of the project that owns the resource. Only administrative and users with advsvc role can specify a project ID other than their own. You cannot change this value through authorization policies.
vlan_transparent (Optional)	body	boolean	Indicates the VLAN transparency mode of the network, which is VLAN transparent (`true`) or not VLAN transparent (`false`).
description (Optional)	body	string	A human-readable description for the resource. Default is an empty string.
is_default (Optional)	body	boolean	The network is default or not.
availability_zone_hints (Optional)	body	array	The availability zone candidate for the network.

Request Example¶

```
{
    "network": {
        "name": "sample_network",
        "admin_state_up": true,
        "dns_domain": "my-domain.org.",
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "mtu": 1400
    }
}
```

Request Example (admin user; single segment mapping)¶

```
{
    "network": {
        "admin_state_up": true,
        "name": "net1",
        "provider:network_type": "vlan",
        "provider:physical_network": "public",
        "provider:segmentation_id": 2,
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e"
    }
}
```

Request Example (admin user; multiple segment mappings)¶

```
{
    "network": {
        "segments": [
            {
                "provider:segmentation_id": 2,
                "provider:physical_network": "public",
                "provider:network_type": "vlan"
            },
            {
                "provider:physical_network": "default",
                "provider:network_type": "flat"
            }
        ],
        "name": "net1",
        "admin_state_up": true,
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e"
    }
}
```

Response Parameters¶

Name	In	Type	Description
network	body	object	A `network` object.
admin_state_up	body	boolean	The administrative state of the network, which is up (`true`) or down (`false`).
availability_zone_hints	body	array	The availability zone candidate for the network.
availability_zones	body	array	The availability zone for the network.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
dns_domain	body	string	A valid DNS domain.
id	body	string	The ID of the network.
ipv4_address_scope	body	string	The ID of the IPv4 address scope that the network is associated with.
ipv6_address_scope	body	string	The ID of the IPv6 address scope that the network is associated with.
l2_adjacency	body	boolean	Indicates whether L2 connectivity is available throughout the `network` .
mtu	body	integer	The maximum transmission unit (MTU) value to address fragmentation. Minimum value is 68 for IPv4, and 1280 for IPv6.
name	body	string	Human-readable name of the network.
port_security_enabled	body	boolean	The port security status of the network. Valid values are enabled (`true`) and disabled (`false`). This value is used as the default value of `port_security_enabled` field of a newly created port.
project_id	body	string	The ID of the project.
provider:network_type	body	string	The type of physical network that this network is mapped to. For example, `flat` , `vlan` , `vxlan` , or `gre` . Valid values depend on a networking back-end.
provider:physical_network	body	string	The physical network where this network/segment is implemented.
provider:segmentation_id	body	integer	The ID of the isolated segment on the physical network. The `network_type` attribute defines the segmentation model. For example, if the `network_type` value is vlan, this ID is a vlan identifier. If the `network_type` value is gre, this ID is a gre key.
qos_policy_id	body	string	The ID of the QoS policy associated with the network.
revision_number	body	integer	The revision number of the resource.
router:external	body	boolean	Indicates whether the network has an external routing facility that’s not managed by the networking service.
segments	body	array	A list of provider `segment` objects.
shared	body	boolean	Indicates whether this network is shared across all tenants. By default, only administrative users can change this value.
status	body	string	The network status. Values are `ACTIVE` , `DOWN` , `BUILD` or `ERROR` .
subnets	body	array	The associated subnets.
tenant_id	body	string	The ID of the project.
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
vlan_transparent	body	boolean	Indicates the VLAN transparency mode of the network, which is VLAN transparent (`true`) or not VLAN transparent (`false`).
description	body	string	A human-readable description for the resource.
is_default	body	boolean	The network is default pool or not.
tags	body	array	The list of tags on the resource.

Response Example¶

```
{
    "network": {
        "admin_state_up": true,
        "availability_zone_hints": [],
        "availability_zones": [
            "nova"
        ],
        "created_at": "2016-03-08T20:19:41",
        "dns_domain": "my-domain.org.",
        "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c",
        "ipv4_address_scope": null,
        "ipv6_address_scope": null,
        "l2_adjacency": true,
        "mtu": 1400,
        "name": "net1",
        "port_security_enabled": true,
        "project_id": "9bacb3c5d39d41a79512987f338cf177",
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "revision_number": 1,
        "router:external": false,
        "shared": false,
        "status": "ACTIVE",
        "subnets": [],
        "tags": ["tag1,tag2"],
        "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
        "updated_at": "2016-03-08T20:19:41",
        "vlan_transparent": false,
        "description": "",
        "is_default": false
    }
}
```

Response Example (admin user; single segment mapping)¶

```
{
    "network": {
        "status": "ACTIVE",
        "subnets": [],
        "availability_zone_hints": [],
        "availability_zones": [
            "nova"
        ],
        "created_at": "2016-03-08T20:19:41",
        "dns_domain": "",
        "ipv4_address_scope": null,
        "ipv6_address_scope": null,
        "name": "net1",
        "provider:physical_network": "public",
        "admin_state_up": true,
        "project_id": "9bacb3c5d39d41a79512987f338cf177",
        "tags": ["tag1,tag2"],
        "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
        "updated_at": "2016-03-08T20:19:41",
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "revision_number": 1,
        "router:external": false,
        "provider:network_type": "vlan",
        "l2_adjacency": true,
        "mtu": 1500,
        "shared": false,
        "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c",
        "provider:segmentation_id": 2,
        "description": "",
        "port_security_enabled": true,
        "is_default": false
    }
}
```

Response Example (admin user; multiple segment mappings)¶

```
{
    "network": {
        "status": "ACTIVE",
        "subnets": [],
        "availability_zone_hints": [],
        "availability_zones": [
            "nova"
        ],
        "created_at": "2016-03-08T20:19:41",
        "name": "net1",
        "admin_state_up": true,
        "dns_domain": "",
        "ipv4_address_scope": null,
        "ipv6_address_scope": null,
        "l2_adjacency": true,
        "mtu": 1500,
        "port_security_enabled": true,
        "project_id": "9bacb3c5d39d41a79512987f338cf177",
        "tags": ["tag1,tag2"],
        "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
        "updated_at": "2016-03-08T20:19:41",
        "qos_policy_id": "6a8454ade84346f59e8d40665f878b2e",
        "revision_number": 1,
        "segments": [
            {
                "provider:segmentation_id": 2,
                "provider:physical_network": "public",
                "provider:network_type": "vlan"
            },
            {
                "provider:segmentation_id": null,
                "provider:physical_network": "default",
                "provider:network_type": "flat"
            }
        ],
        "shared": false,
        "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c",
        "description": "",
        "is_default": false
    }
}
```

Ports¶

Lists, shows details for, creates, updates, and deletes ports.

Allowed address pairs¶

The `allowed-address-pairs` extension adds an `allowed_address_pairs` attribute to ports. The value of `allowed_address_pairs` is an array of allowed address pair objects, each having an `ip_address` and a `mac_address`. The set of allowed address pairs defines IP and MAC address that the port can use when sending packets if `port_security_enabled` is `true` (see the `port-security` extension). Note that while the `ip_address` is required in each allowed address pair, the `mac_address` is optional and will be taken from the port if not specified.

Data plane status extension¶

The data plane port extension (`data-plane-status`) adds a new attribute `data_plane_status` to represent the status of the underlying data plane. This attribute is to be managed by entities outside of the Networking service, while the `status` attribute is managed by Networking service. Both status attributes are independent from one another.

Supported data plane status values:

- `null`: no status being reported; default value
- `ACTIVE`: the underlying data plane is up and running
- `DOWN`: no traffic can flow from/to the port

DNS integration¶

The `dns-integration` extension adds the `dns_name` and `dns_assignment` attributes to port resources. While the `dns_name` can be set on create and update operations, the `dns_assignment` is read-only and shows the `hostname`, `ip_address` and `fqdn` for the port's internal DNS assignment.

To enable the `dns_domain` on port resources, the `dns-domain-ports` extension must be used in conjunction with the `dns-integration` extension. When enabled and set, a port level `dns_domain` take precedence over a `dns_domain` specified in the port's network allowing per-port DNS domains.

Extra DHCP option (`extra_dhcp_opt`) extension¶

The extra DHCP option (`extra_dhcp_opt`) extension enables extra DHCP configuration options on `ports`. For example, PXE boot options to DHCP clients can be specified (e.g. tftp-server, server-ip-address, bootfile-name). The value of the `extra_dhcp_opt` attribute is an array of DHCP option objects, where each object contains an `opt_name` and `opt_value` (string values) as well as an optional `ip_version` (the acceptable values are either the integer `4` or `6`).

IP allocation extension¶

The IP allocation extension (`ip_allocation`) adds a new read-only attribute `ip_allocation` that indicates when ports use deferred, immediate or no IP allocation.

IP Substring Filtering¶

The `ip-substring-filtering` extension adds support for filtering ports by using part of an IP address.

Mac learning extension¶

The `mac_learning_enabled` extension extends neutron ports providing the ability to enable MAC learning on the associated port via the `mac_learning_enabled` attribute.

Port binding extended attributes¶

The port binding extension (`binding`) allows administrative users to specify and retrieve physical binding information of ports. The extension defines several attributes whose names have a prefix `binding:` including `binding:host_id`, `binding:vnic_type`, `binding:vif_type`, `binding:vif_details`, and `binding:profile`.

Port resource request¶

The port resource request extension (`port-resource-request`) allows administrative users (including Nova) to retrieve the Placement resources and traits needed by a port by introducing the `resource_request` to `port` resources.

Port security¶

The `port-security` extension adds the `port_security_enabled` boolean attribute to ports. If a `port-security` value is not specified during port creation, a port will inherit the `port_security_enabled` from the network its connected to.

QoS extension¶

The QoS extension (`qos`) makes it possible to define QoS policies and associate these to the ports by introducing the `qos_policy_id` attribute. The policies should be created before they are associated to the ports.

Regenerate mac address extension¶

The Port MAC address regenerate extension (`port-mac-address-regenerate`) makes it possible to regenerate the mac address of a port. When passing `'null'` (`None`) as the `mac_address` on port update, a new mac address will be generated and set on the port.

Resource timestamps¶

The `standard-attr-timestamp` extension adds the `created_at` and `updated_at` attributes to all resources that have standard attributes.

Tag extension¶

The `standard-attr-tag` adds Tag support for resources with standard attributes by adding the `tags` attribute allowing consumers to associate tags with resources.

Uplink status propagation¶

The `uplink-status-propagation` extension adds `uplink_status_propagation` attribute to port. If this attribute is set to `true`, uplink status propagation is enabled. If this attribute is not specified, it is default to `false` which indicates uplink status propagation is disabled.

Show port details GET /v2.0/ports/{port_id}

Shows details for a port.

Use the `fields` query parameter to control which fields are returned in the response body. For information, see Filtering and Column Selection.

Normal response codes: 200

Error response codes: 401, 404

Request¶

Name	In	Type	Description
port_id	path	string	The ID of the port.
fields (Optional)	query	string	The fields that you want the server to return. If no `fields` query parameter is specified, the networking API returns all attributes allowed by the policy settings. By using `fields` parameter, the API returns only the requested set of attributes. `fields` parameter can be specified multiple times. For example, if you specify `fields=id&fields=name` in the request URL, only `id` and `name` attributes will be returned.

Response Parameters¶

Name	In	Type	Description
port	body	object	A `port` object.
admin_state_up	body	boolean	The administrative state of the resource, which is up (`true`) or down (`false`).
allowed_address_pairs	body	array	A set of zero or more allowed address pair objects each where address pair object contains an `ip_address` and `mac_address`. While the `ip_address` is required, the `mac_address` will be taken from the port if not specified. The value of `ip_address` can be an IP Address or a CIDR (if supported by the underlying extension plugin). A server connected to the port can send a packet with source address which matches one of the specified allowed address pairs.
binding:host_id	body	string	The ID of the host where the port resides.
binding:profile	body	object	A dictionary that enables the application running on the specific host to pass and receive vif port information specific to the networking back-end. The networking API does not define a specific format of this field.
binding:vif_details	body	object	A dictionary which contains additional information on the port. Currently the following fields are defined: `port_filter` and `ovs_hybrid_plug`. `port_filter` is a boolean indicating the networking service provides port filtering features such as security group and/or anti MAC/IP spoofing. `ovs_hybrid_plug` is a boolean used to inform an API consumer like nova that the hybrid plugging strategy for OVS should be used.

binding:vif_type	body	string	The type of which mechanism is used for the port. An API consumer like nova can use this to determine an appropriate way to attach a device (for example an interface of a virtual server) to the port. Available values currently defined includes `ovs`, `bridge`, `macvtap`, `hw_veb`, `hostdev_physical`, `vhostuser`, `distributed` and `other`. There are also special values: `unbound` and `binding_failed`. `unbound` means the port is not bound to a networking back-end. `binding_failed` means an error that the port failed to be bound to a networking back-end.
binding:vnic_type	body	string	The type of vNIC which this port should be attached to. This is used to determine which mechanism driver(s) to be used to bind the port. The valid values are `normal`, `macvtap`, `direct`, `baremetal`, `direct-physical` and `virtio-forwarder`. What type of vNIC is actually available depends on deployments.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
data_plane_status	body	string	Status of the underlying data plane of a port.
description	body	string	A human-readable description for the resource.
device_id	body	string	The ID of the device that uses this port. For example, a server instance or a logical router.
device_owner	body	string	The entity type that uses this port. For example, `compute:nova` (server instance), `network:dhcp` (DHCP agent) or `network:router_interface` (router interface).
dns_assignment	body	object	Data assigned to a port by the Networking internal DNS including the `hostname`, `ip_address` and `fqdn`.
dns_domain	body	string	A valid DNS domain.
dns_name	body	string	A valid DNS name.
extra_dhcp_opts	body	array	A set of zero or more extra DHCP option pairs. An option pair consists of an option value and name.
fixed_ips	body	array	The IP addresses for the port. If the port has multiple IP addresses, this field has multiple entries. Each entry consists of IP address (`ip_address`) and the subnet ID from which the IP address is assigned (`subnet_id`).
id	body	string	The ID of the resource.
ip_allocation	body	string	Indicates when ports use either `deferred`, `immediate` or no IP allocation (`none`).
mac_address	body	string	The MAC address of the port.
name	body	string	Human-readable name of the resource.
network_id	body	string	The ID of the attached network.
port_security_enabled	body	boolean	The port security status. A valid value is enabled (`true`) or disabled (`false`). If port security is enabled for the port, security group rules and anti-spoofing rules are applied to the traffic on the port. If disabled, no such rules are applied.
project_id	body	string	The ID of the project.
revision_number	body	integer	The revision number of the resource.
qos_policy_id	body	string	The ID of the QoS policy associated with the port.
resource_request (Optional)	body	object	Expose Placement resources (i.e.: `minimum-bandwidth`) and traits (i.e.: `vnic-type`, `physnet`) requested by a port to Nova and Placement. A `resource_request` object contains a `required` key for the traits (generated from the `vnic_type` and the `physnet`) required by the port, and a `resources` key for `ingress` and `egress` `minimum-bandwidth` need for the port.
security_groups	body	array	The IDs of security groups applied to the port.
status	body	string	The port status. Values are `ACTIVE`, `DOWN`, `BUILD` and `ERROR`.
tags	body	array	The list of tags on the resource.
tenant_id	body	string	The ID of the project.
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
uplink_status_propagation	body	boolean	The uplink status propagation of the port. Valid values are enabled (`true`) and disabled (`false`).
mac_learning_enabled (Optional)	body	boolean	A boolean value that indicates if MAC Learning is enabled on the associated port.

Response Example¶

```
{
    "port": {
        "admin_state_up": true,
        "allowed_address_pairs": [],
        "created_at": "2016-03-08T20:19:41",
        "data_plane_status": "ACTIVE",
        "description": "",
        "device_id": "5e3898d7-11be-483e-9732-b2f5eccd2b2e",
        "device_owner": "network:router_interface",
        "dns_assignment": {
            "hostname": "myport",
            "ip_address": "10.0.0.1",
            "fqdn": "myport.my-domain.org"
        },
        "dns_domain": "my-domain.org.",
        "dns_name": "myport",
        "extra_dhcp_opts": [
            {
                "opt_value": "pxelinux.0",
                "ip_version": 4,
                "opt_name": "bootfile-name"
            }
        ],
        "fixed_ips": [
            {
                "ip_address": "10.0.0.1",
                "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2"
            }
        ],
        "id": "46d4bfb9-b26e-41f3-bd2e-e6dcc1ccedb2",
        "ip_allocation": "immediate",
        "mac_address": "fa:16:3e:23:fd:d7",
        "name": "",
        "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
        "port_security_enabled": false,
        "project_id": "7e02058126cc4950b75f9970368ba177",
        "revision_number": 1,
        "security_groups": [],
        "status": "ACTIVE",
        "tags": ["tag1,tag2"],
        "tenant_id": "7e02058126cc4950b75f9970368ba177",
        "updated_at": "2016-03-08T20:19:41",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
        "uplink_status_propagation": false
    }
}
```

Response Example (admin user)¶

```
{
    "port": {
        "admin_state_up": true,
        "allowed_address_pairs": [],
        "binding:host_id": "devstack",
        "binding:profile": {},
        "binding:vif_details": {
            "ovs_hybrid_plug": true,
            "port_filter": true
        },
        "binding:vif_type": "ovs",
        "binding:vnic_type": "normal",
        "created_at": "2016-03-08T20:19:41",
        "data_plane_status": "ACTIVE",
        "description": "",
        "device_id": "5e3898d7-11be-483e-9732-b2f5eccd2b2e",
        "device_owner": "network:router_interface",
        "dns_assignment": {
            "hostname": "myport",
            "ip_address": "10.0.0.1",
            "fqdn": "myport.my-domain.org"
        },
        "dns_domain": "my-domain.org.",
        "dns_name": "myport",
        "extra_dhcp_opts": [
            {
                "opt_value": "pxelinux.0",
                "ip_version": 4,
                "opt_name": "bootfile-name"
            }
        ],
        "fixed_ips": [
            {
                "ip_address": "10.0.0.1",
                "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2"
            }
        ],
        "id": "46d4bfb9-b26e-41f3-bd2e-e6dcc1ccedb2",
        "ip_allocation": "immediate",
        "mac_address": "fa:16:3e:23:fd:d7",
        "mac_learning_enabled": false,
        "name": "",
        "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
        "port_security_enabled": false,
        "project_id": "7e02058126cc4950b75f9970368ba177",
        "revision_number": 1,
        "security_groups": [],
        "status": "ACTIVE",
        "tags": ["tag1,tag2"],
        "tenant_id": "7e02058126cc4950b75f9970368ba177",
        "updated_at": "2016-03-08T20:19:41",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
        "resource_request": {
            "required": ["CUSTOM_PHYSNET_PUBLIC", "CUSTOM_VNIC_TYPE_NORMAL"],
            "resources": {"NET_BW_EGR_KILOBIT_PER_SEC": 1000}
        },
        "uplink_status_propagation": false
    }
}
```

Update port PUT /v2.0/ports/{port_id}

Updates a port.

You can update information for a port, such as its symbolic name and associated IPs. When you update IPs for a port, any previously associated IPs are removed, returned to the respective subnet allocation pools, and replaced by the IPs in the request body. Therefore, this operation replaces the `fixed_ip` attribute when you specify it in the request body. If the updated IP addresses are not valid or are already in use, the operation fails and the existing IP addresses are not removed from the port.

When you update security groups for a port and the operation succeeds, any associated security groups are removed and replaced by the security groups in the request body. Therefore, this operation replaces the `security_groups` attribute when you specify it in the request body. If the security groups are not valid, the operation fails and the existing security groups are not removed from the port.

Only admins and users with a specific role can update the data plane status (default role: `data_plane_integrator`).

Normal response codes: 200

Error response codes: 400, 401, 403, 404, 409, 412

Request¶

Name	In	Type	Description
port_id	path	string	The ID of the port.
port	body	object	A `port` object.
admin_state_up (Optional)	body	boolean	The administrative state of the resource, which is up (`true`) or down (`false`). Default is `true`.
allowed_address_pairs (Optional)	body	array	A set of zero or more allowed address pair objects each where address pair object contains an `ip_address` and `mac_address`. While the `ip_address` is required, the `mac_address` will be taken from the port if not specified. The value of `ip_address` can be an IP Address or a CIDR (if supported by the underlying extension plugin). A server connected to the port can send a packet with source address which matches one of the specified allowed address pairs.
binding:host_id (Optional)	body	string	The ID of the host where the port resides. The default is an empty string.
binding:profile (Optional)	body	object	A dictionary that enables the application running on the specific host to pass and receive vif port information specific to the networking back-end. The networking API does not define a specific format of this field. The default is an empty dictionary.
binding:vnic_type (Optional)	body	string	The type of vNIC which this port should be attached to. This is used to determine which mechanism driver(s) to be used to bind the port. The valid values are `normal`, `macvtap`, `direct`, `baremetal`, `direct-physical` and `virtio-forwarder`. What type of vNIC is actually available depends on deployments. The default is `normal`.
data_plane_status (Optional)	body	string	Status of the underlying data plane of a port.
description (Optional)	body	string	A human-readable description for the resource. Default is an empty string.
device_id (Optional)	body	string	The ID of the device that uses this port. For example, a server instance or a logical router.
device_owner (Optional)	body	string	The entity type that uses this port. For example, `compute:nova` (server instance), `network:dhcp` (DHCP agent) or `network:router_interface` (router interface).
dns_domain (Optional)	body	string	A valid DNS domain.
dns_name (Optional)	body	string	A valid DNS name.
extra_dhcp_opts (Optional)	body	array	A set of zero or more extra DHCP option pairs. An option pair consists of an option value and name.
fixed_ips (Optional)	body	array	The IP addresses for the port. If you would like to assign multiple IP addresses for the port, specify multiple entries in this field. Each entry consists of IP address (`ip_address`) and the subnet ID from which the IP address is assigned (`subnet_id`).If you specify both a subnet ID and an IP address, OpenStack Networking tries to allocate the IP address on that subnet to the port.If you specify only a subnet ID, OpenStack Networking allocates an available IP from that subnet to the port.If you specify only an IP address, OpenStack Networking tries to allocate the IP address if the address is a valid IP for any of the subnets on the specified network.
mac_address (Optional)	body	string	The MAC address of the port. By default, only administrative users and users with advsvc role can change this value.
name (Optional)	body	string	Human-readable name of the resource. Default is an empty string.
port_security_enabled (Optional)	body	boolean	The port security status. A valid value is enabled (`true`) or disabled (`false`). If port security is enabled for the port, security group rules and anti-spoofing rules are applied to the traffic on the port. If disabled, no such rules are applied.
qos_policy_id (Optional)	body	string	QoS policy associated with the port.
security_groups (Optional)	body	array	The IDs of security groups applied to the port.
mac_learning_enabled (Optional)	body	boolean	A boolean value that indicates if MAC Learning is enabled on the associated port.

Request Example¶

```
{
    "port": {
        "admin_state_up": true,
        "device_id": "d90a13da-be41-461f-9f99-1dbcf438fdf2",
        "device_owner": "compute:nova",
        "name": "test-for-port-update",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae"
    }
}
```

Request Example (admin user)¶

```
{
    "port": {
        "binding:host_id": "test_for_port_update_host",
        "device_id": "d90a13da-be41-461f-9f99-1dbcf438fdf2",
        "data_plane_status": "DOWN",
        "device_owner": "compute:nova",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae"
    }
}
```

Response Parameters¶

Name	In	Type	Description
port	body	object	A `port` object.
admin_state_up	body	boolean	The administrative state of the resource, which is up (`true`) or down (`false`).
allowed_address_pairs	body	array	A set of zero or more allowed address pair objects each where address pair object contains an `ip_address` and `mac_address` . While the `ip_address` is required, the `mac_address` will be taken from the port if not specified. The value of `ip_address` can be an IP Address or a CIDR (if supported by the underlying extension plugin). A server connected to the port can send a packet with source address which matches one of the specified allowed address pairs.
binding:host_id	body	string	The ID of the host where the port resides.
binding:profile	body	object	A dictionary that enables the application running on the specific host to pass and receive vif port information specific to the networking back-end. The networking API does not define a specific format of this field.
binding:vif_details	body	object	A dictionary which contains additional information on the port. Currently the following fields are defined: `port_filter` and `ovs_hybrid_plug` . `port_filter` is a boolean indicating the networking service provides port filtering features such as security group and/or anti MAC/IP spoofing. `ovs_hybrid_plug` is a boolean used to inform an API consumer like nova that the hybrid plugging strategy for OVS should be used.
binding:vif_type	body	string	The type of which mechanism is used for the port. An API consumer like nova can use this to determine an appropriate way to attach a device (for example an interface of a virtual server) to the port. Available values currently defined includes `ovs` , `bridge` , `macvtap` , `hw_veb` , `hostdev_physical` , `vhostuser` , `distributed` and `other` . There are also special values: `unbound` and `binding_failed` . `unbound` means the port is not bound to a networking back-end. `binding_failed` means an error that the port failed to be bound to a networking back-end.
binding:vnic_type	body	string	The type of vNIC which this port should be attached to. This is used to determine which mechanism driver(s) to be used to bind the port. The valid values are `normal` , `macvtap` , `direct` , `baremetal` , `direct-physical` and `virtio-forwarder` . What type of vNIC is actually available depends on deployments.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
data_plane_status	body	string	Status of the underlying data plane of a port.
description	body	string	A human-readable description for the resource.
device_id	body	string	The ID of the device that uses this port. For example, a server instance or a logical router.
device_owner	body	string	The entity type that uses this port. For example, `compute:nova` (server instance), `network:dhcp` (DHCP agent) or `network:router_interface` (router interface).
dns_assignment	body	object	Data assigned to a port by the Networking internal DNS including the `hostname` , `ip_address` and `fqdn` .
dns_domain	body	string	A valid DNS domain.
dns_name	body	string	A valid DNS name.

extra_dhcp_opts	body	array	A set of zero or more extra DHCP option pairs. An option pair consists of an option value and name.
fixed_ips	body	array	The IP addresses for the port. If the port has multiple IP addresses, this field has multiple entries. Each entry consists of IP address (`ip_address`) and the subnet ID from which the IP address is assigned (`subnet_id`).
id	body	string	The ID of the resource.
ip_allocation	body	string	Indicates when ports use either `deferred` , `immediate` or no IP allocation (`none`).
mac_address	body	string	The MAC address of the port.
name	body	string	Human-readable name of the resource.
network_id	body	string	The ID of the attached network.
port_security_enabled	body	boolean	The port security status. A valid value is enabled (`true`) or disabled (`false`). If port security is enabled for the port, security group rules and anti-spoofing rules are applied to the traffic on the port. If disabled, no such rules are applied.
project_id	body	string	The ID of the project.
revision_number	body	integer	The revision number of the resource.
qos_policy_id	body	string	The ID of the QoS policy associated with the port.
resource_request (Optional)	body	object	Expose Placement resources (i.e.: `minimum-bandwidth`) and traits (i.e.: `vnic-type` , `physnet`) requested by a port to Nova and Placement. A `resource_request` object contains a `required` key for the traits (generated from the `vnic_type` and the `physnet`) required by the port, and a `resources` key for `ingress` and `egress` `minimum-bandwidth` need for the port.
security_groups	body	array	The IDs of security groups applied to the port.
status	body	string	The port status. Values are `ACTIVE` , `DOWN` , `BUILD` and `ERROR` .
tags	body	array	The list of tags on the resource.
tenant_id	body	string	The ID of the project.
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
uplink_status_propagation	body	boolean	The uplink status propagation of the port. Valid values are enabled (`true`) and disabled (`false`).
mac_learning_enabled (Optional)	body	boolean	A boolean value that indicates if MAC Learning is enabled on the associated port.

Response Example¶

```
{
    "port": {
        "admin_state_up": true,
        "allowed_address_pairs": [],
        "binding:host_id": "test_for_port_update_host",
        "binding:profile": {},
        "binding:vif_details": {},
        "binding:vif_type": "binding_failed",
        "binding:vnic_type": "normal",
        "created_at": "2016-03-08T20:19:41",
        "data_plane_status": "ACTIVE",
        "description": "",
        "device_id": "d90a13da-be41-461f-9f99-1dbcf438fdf2",
        "device_owner": "compute:nova",
        "dns_assignment": {
            "hostname": "myport",
            "ip_address": "20.20.0.4",
            "fqdn": "myport.my-domain.org"
        },
        "dns_domain": "my-domain.org.",
        "dns_name": "myport",
        "extra_dhcp_opts": [
            {
                "opt_value": "pxelinux.0",
                "ip_version": 4,
                "opt_name": "bootfile-name"
            }
        ],
        "fixed_ips": [
            {
                "ip_address": "20.20.0.4",
                "subnet_id": "898dec4a-74df-4193-985f-c76721bcc746"
            }
        ],
        "id": "43c831e0-19ce-4a76-9a49-57b57e69428b",
        "ip_allocation": "immediate",
        "mac_address": "fa:16:3e:11:11:5e",
        "name": "test-for-port-update",
        "network_id": "883fc383-5ea1-4c8b-8916-e1ddb0a9f365",
        "project_id": "522eda8d23124b25bf03fe44f1986b74",
        "revision_number": 1,
        "security_groups": [
            "ce0179d6-8a94-4f7c-91c2-f3038e2acbd0"
        ],
        "status": "DOWN",
        "tags": ["tag1,tag2"],
        "tenant_id": "522eda8d23124b25bf03fe44f1986b74",
        "updated_at": "2016-03-08T20:19:41",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
        "port_security_enabled": false,
        "uplink_status_propagation": false
    }
}
```

Response Example (admin user)¶

```
{
    "port": {
        "admin_state_up": true,
        "allowed_address_pairs": [],
        "binding:host_id": "test_for_port_update_host",
        "binding:profile": {},
        "binding:vif_details": {},
        "binding:vif_type": "binding_failed",
        "binding:vnic_type": "normal",
        "created_at": "2016-03-08T20:19:41",
        "data_plane_status": "DOWN",
        "description": "",
        "device_id": "d90a13da-be41-461f-9f99-1dbcf438fdf2",
        "device_owner": "compute:nova",
        "dns_assignment": {
            "hostname": "myport",
            "ip_address": "20.20.0.4",
            "fqdn": "myport.my-domain.org"
        },
        "dns_domain": "my-domain.org.",
        "dns_name": "myport",
        "extra_dhcp_opts": [
            {
                "opt_value": "pxelinux.0",
                "ip_version": 4,
                "opt_name": "bootfile-name"
            }
        ],
        "fixed_ips": [
            {
                "ip_address": "20.20.0.4",
                "subnet_id": "898dec4a-74df-4193-985f-c76721bcc746"
            }
        ],
        "id": "43c831e0-19ce-4a76-9a49-57b57e69428b",
        "ip_allocation": "immediate",
        "mac_address": "fa:16:3e:11:11:5e",
        "name": "test-for-port-update",
        "network_id": "883fc383-5ea1-4c8b-8916-e1ddb0a9f365",
        "project_id": "522eda8d23124b25bf03fe44f1986b74",
        "revision_number": 2,
        "security_groups": [
            "ce0179d6-8a94-4f7c-91c2-f3038e2acbd0"
        ],
        "status": "DOWN",
        "tags": ["tag1,tag2"],
        "tenant_id": "522eda8d23124b25bf03fe44f1986b74",
        "updated_at": "2016-03-08T20:19:41",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
        "port_security_enabled": false,
        "resource_request": {
            "required": ["CUSTOM_PHYSNET_PUBLIC", "CUSTOM_VNIC_TYPE_NORMAL"],
            "resources": {"NET_BW_EGR_KILOBIT_PER_SEC": 1000}
        },
        "uplink_status_propagation": false
    }
}
```

Delete port DELETE /v2.0/ports/{port_id}

Deletes a port.

Any IP addresses that are associated with the port are returned to the respective subnets allocation pools.

Normal response codes: 204

Error response codes: 401, 403, 404, 412

Request¶

Name	In	Type	Description
port_id	path	string	The ID of the port.

Response¶

There is no body content for the response of a successful DELETE request.

List ports GET /v2.0/ports

Lists ports to which the user has access.

Default policy settings return only those ports that are owned by the project of the user who submits the request, unless the request is submitted by a user with administrative rights.

Use the `fields` query parameter to control which fields are returned in the response body. Additionally, you can filter results by using query string parameters. For information, see Filtering and Column Selection.

If the `ip-substring-filtering` extension is enabled, the Neutron API supports IP address substring filtering on the `fixed_ips` attribute. If you specify an IP address substring (`ip_address_substr`) in an entry of the `fixed_ips` attribute, the Neutron API will list all ports that have an IP address matching the substring.

Normal response codes: 200

Error response codes: 401

Request¶

Name	In	Type	Description
admin_state_up (Optional)	query	boolean	Filter the list result by the administrative state of the resource, which is up (`true`) or down (`false`).
binding:host_id (Optional)	query	string	Filter the port list result by the ID of the host where the port resides.
description (Optional)	query	string	Filter the list result by the human-readable description of the resource.
device_id (Optional)	query	string	Filter the port list result by the ID of the device that uses this port. For example, a server instance or a logical router.
device_owner (Optional)	query	string	Filter the port result list by the entity type that uses this port. For example, `compute:nova` (server instance), `network:dhcp` (DHCP agent) or `network:router_interface` (router interface).
fixed_ips (Optional)	query	array	Filter the port list result by the IP addresses for the port. This field has one or multiple entries. Each entry consists of IP address (`ip_address`), IP address substring (`ip_address_substr`) and/or the subnet ID from which the IP address is assigned (`subnet_id`).
id (Optional)	query	string	Filter the list result by the ID of the resource.
ip_allocation (Optional)	query	string	Filter the port list result based on if the ports use `deferred`, `immediate` or no IP allocation (`none`).
mac_address (Optional)	query	string	Filter the port list result by the MAC address of the port.
name (Optional)	query	string	Filter the list result by the human-readable name of the resource.
network_id (Optional)	query	string	Filter the list result by the ID of the attached network.
project_id (Optional)	query	string	Filter the list result by the ID of the project that owns the resource.
revision_number (Optional)	query	integer	Filter the list result by the revision number of the resource.
sort_dir (Optional)	query	string	Sort direction. A valid value is `asc` (ascending) or `desc` (descending). You can specify multiple pairs of sort key and sort direction query parameters.
sort_key (Optional)	query	string	Sorts by a port attribute. You can specify multiple pairs of sort key and sort direction query parameters. The sort keys are limited to:admin_state_up``device_id``device_owner``id``ip_allocation``mac_address``name``network_id``project_id``status``tenant_id
status (Optional)	query	string	Filter the port list result by the port status. Values are `ACTIVE`, `DOWN`, `BUILD` and `ERROR`.
tenant_id (Optional)	query	string	Filter the list result by the ID of the project that owns the resource.
tags (Optional)	query	string	A list of tags to filter the list result by. Resources that match all tags in this list will be returned. Tags in query must be separated by comma.
tags-any (Optional)	query	string	A list of tags to filter the list result by. Resources that match any tag in this list will be returned. Tags in query must be separated by comma.
not-tags (Optional)	query	string	A list of tags to filter the list result by. Resources that match all tags in this list will be excluded. Tags in query must be separated by comma.
not-tags-any (Optional)	query	string	A list of tags to filter the list result by. Resources that match any tag in this list will be excluded. Tags in query must be separated by comma.
fields (Optional)	query	string	The fields that you want the server to return. If no `fields` query parameter is specified, the networking API returns all attributes allowed by the policy settings. By using `fields` parameter, the API returns only the requested set of attributes. `fields` parameter can be specified multiple times. For example, if you specify `fields=id&fields=name` in the request URL, only `id` and `name` attributes will be returned.
mac_learning_enabled (Optional)	query	boolean	Filter the list result by the mac_learning_enabled state of the resource, which is enabled (`true`) or disabled (`false`).

Response Parameters¶

Name	In	Type	Description
ports	body	array	A list of `port` objects.
admin_state_up	body	boolean	The administrative state of the resource, which is up (`true`) or down (`false`).
allowed_address_pairs	body	array	A set of zero or more allowed address pair objects each where address pair object contains an `ip_address` and `mac_address`. While the `ip_address` is required, the `mac_address` will be taken from the port if not specified. The value of `ip_address` can be an IP Address or a CIDR (if supported by the underlying extension plugin). A server connected to the port can send a packet with source address which matches one of the specified allowed address pairs.
binding:host_id	body	string	The ID of the host where the port resides.
binding:profile	body	object	A dictionary that enables the application running on the specific host to pass and receive vif port information specific to the networking back-end. The networking API does not define a specific format of this field.
binding:vif_details	body	object	A dictionary which contains additional information on the port. Currently the following fields are defined: `port_filter` and `ovs_hybrid_plug`. `port_filter` is a boolean indicating the networking service provides port filtering features such as security group and/or anti MAC/IP spoofing. `ovs_hybrid_plug` is a boolean used to inform an API consumer like nova that the hybrid plugging strategy for OVS should be used.
binding:vif_type	body	string	The type of which mechanism is used for the port. An API consumer like nova can use this to determine an appropriate way to attach a device (for example an interface of a virtual server) to the port. Available values currently defined includes `ovs`, `bridge`, `macvtap`, `hw_veb`, `hostdev_physical`, `vhostuser`, `distributed` and `other`. There are also special values: `unbound` and `binding_failed`. `unbound` means the port is not bound to a networking back-end. `binding_failed` means an error that the port failed to be bound to a networking back-end.
binding:vnic_type	body	string	The type of vNIC which this port should be attached to. This is used to determine which mechanism driver(s) to be used to bind the port. The valid values are `normal`, `macvtap`, `direct`, `baremetal`, `direct-physical` and `virtio-forwarder`. What type of vNIC is actually available depends on deployments.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).

data_plane_status	body	string	Status of the underlying data plane of a port.
description	body	string	A human-readable description for the resource.
device_id	body	string	The ID of the device that uses this port. For example, a server instance or a logical router.
device_owner	body	string	The entity type that uses this port. For example, `compute:nova` (server instance), `network:dhcp` (DHCP agent) or `network:router_interface` (router interface).
dns_assignment	body	object	Data assigned to a port by the Networking internal DNS including the `hostname`, `ip_address` and `fqdn`.
dns_domain	body	string	A valid DNS domain.
dns_name	body	string	A valid DNS name.
extra_dhcp_opts	body	array	A set of zero or more extra DHCP option pairs. An option pair consists of an option value and name.
fixed_ips	body	array	The IP addresses for the port. If the port has multiple IP addresses, this field has multiple entries. Each entry consists of IP address (`ip_address`) and the subnet ID from which the IP address is assigned (`subnet_id`).
id	body	string	The ID of the resource.
ip_allocation	body	string	Indicates when ports use either `deferred`, `immediate` or no IP allocation (`none`).
mac_address	body	string	The MAC address of the port.
name	body	string	Human-readable name of the resource.
network_id	body	string	The ID of the attached network.
port_security_enabled	body	boolean	The port security status. A valid value is enabled (`true`) or disabled (`false`). If port security is enabled for the port, security group rules and anti-spoofing rules are applied to the traffic on the port. If disabled, no such rules are applied.
project_id	body	string	The ID of the project.
revision_number	body	integer	The revision number of the resource.
qos_policy_id	body	string	The ID of the QoS policy associated with the port.
resource_request (Optional)	body	object	Expose Placement resources (i.e.: `minimum-bandwidth`) and traits (i.e.: `vnic-type`, `physnet`) requested by a port to Nova and Placement. A `resource_request` object contains a `required` key for the traits (generated from the `vnic_type` and the `physnet`) required by the port, and a `resources` key for `ingress` and `egress` `minimum-bandwidth` need for the port.
security_groups	body	array	The IDs of security groups applied to the port.
status	body	string	The port status. Values are `ACTIVE`, `DOWN`, `BUILD` and `ERROR`.
tags	body	array	The list of tags on the resource.
tenant_id	body	string	The ID of the project.
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
uplink_status_propagation	body	boolean	The uplink status propagation of the port. Valid values are enabled (`true`) and disabled (`false`).
mac_learning_enabled (Optional)	body	boolean	A boolean value that indicates if MAC Learning is enabled on the associated port.

Response Example¶

```
{
    "ports": [
        {
            "admin_state_up": true,
            "allowed_address_pairs": [],
            "created_at": "2016-03-08T20:19:41",
            "data_plane_status": null,
            "description": "",
            "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824",
            "device_owner": "network:router_gateway",
            "dns_assignment": {
                "hostname": "myport",
                "ip_address": "172.24.4.2",
                "fqdn": "myport.my-domain.org"
            },
            "dns_domain": "my-domain.org.",
            "dns_name": "myport",
            "extra_dhcp_opts": [
            {
                "opt_value": "pxelinux.0",
                "ip_version": 4,
                "opt_name": "bootfile-name"
            }
            ],
            "fixed_ips": [
                {
                    "ip_address": "172.24.4.2",
                    "subnet_id": "008ba151-0b8c-4a67-98b5-0d2b87666062"
                }
            ],
            "id": "d80b1a3b-4fc1-49f3-952e-1e2ab7081d8b",
            "ip_allocation": "immediate",
            "mac_address": "fa:16:3e:58:42:ed",
            "name": "",
            "network_id": "70c1db1f-b701-45bd-96e0-a313ee3430b3",
            "project_id": "",
            "revision_number": 1,
            "security_groups": [],
            "status": "ACTIVE",
            "tags": ["tag1,tag2"],
            "tenant_id": "",
            "updated_at": "2016-03-08T20:19:41",
            "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
            "port_security_enabled": false,
            "uplink_status_propagation": false
        },
        {
            "admin_state_up": true,
            "allowed_address_pairs": [],
            "created_at": "2016-03-08T20:19:41",
            "data_plane_status": null,
            "description": "",
            "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824",
            "device_owner": "network:router_interface",
            "dns_assignment": {
                "hostname": "myport2",
                "ip_address": "10.0.0.1",
                "fqdn": "myport2.my-domain.org"
            },
            "dns_domain": "my-domain.org.",
            "dns_name": "myport2",
            "extra_dhcp_opts": [
                {
                    "opt_value": "pxelinux.0",
                    "ip_version": 4,
                    "opt_name": "bootfile-name"
                }
            ],
            "fixed_ips": [
                {
                    "ip_address": "10.0.0.1",
                    "subnet_id": "288bf4a1-51ba-43b6-9d0a-520e9005db17"
                }
            ],
            "id": "f71a6703-d6de-4be1-a91a-a570ede1d159",
            "ip_allocation": "immediate",
            "mac_address": "fa:16:3e:bb:3c:e4",
            "name": "",
            "network_id": "f27aa545-cbdd-4907-b0c6-c9e8b039dcc2",
            "project_id": "d397de8a63f341818f198abb0966f6f3",
            "revision_number": 1,
            "security_groups": [],
            "status": "ACTIVE",
            "tags": ["tag1,tag2"],
            "tenant_id": "d397de8a63f341818f198abb0966f6f3",
            "updated_at": "2016-03-08T20:19:41",
            "qos_policy_id": null,
            "port_security_enabled": false,
            "uplink_status_propagation": false
        }
    ]
}
```

Response Example (admin user)¶

```
{
    "ports": [
        {
            "admin_state_up": true,
            "allowed_address_pairs": [],
            "binding:host_id": "devstack",
            "binding:profile": {},
            "binding:vif_details": {
                "ovs_hybrid_plug": true,
                "port_filter": true
            },
            "binding:vif_type": "ovs",
            "binding:vnic_type": "normal",
            "created_at": "2016-03-08T20:19:41",
            "data_plane_status": null,
            "description": "",
            "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824",
            "device_owner": "network:router_gateway",
            "dns_assignment": {
                "hostname": "myport",
                "ip_address": "172.24.4.2",
                "fqdn": "myport.my-domain.org"
            },
            "dns_domain": "my-domain.org.",
            "dns_name": "myport",
            "extra_dhcp_opts": [],
            "fixed_ips": [
                {
                    "ip_address": "172.24.4.2",
                    "subnet_id": "008ba151-0b8c-4a67-98b5-0d2b87666062"
                }
            ],
            "id": "d80b1a3b-4fc1-49f3-952e-1e2ab7081d8b",
            "ip_allocation": "immediate",
            "mac_address": "fa:16:3e:58:42:ed",
            "name": "",
            "network_id": "70c1db1f-b701-45bd-96e0-a313ee3430b3",
            "port_security_enabled": true,
            "project_id": "",
            "revision_number": 1,
            "security_groups": [],
            "status": "ACTIVE",
            "tenant_id": "",
            "updated_at": "2016-03-08T20:19:41",
            "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
            "resource_request": {
                "required": ["CUSTOM_PHYSNET_PUBLIC", "CUSTOM_VNIC_TYPE_NORMAL"],
                "resources": {"NET_BW_EGR_KILOBIT_PER_SEC": 1000}
            },
            "tags": ["tag1,tag2"],
            "tenant_id": "",
            "uplink_status_propagation": false
        },
        {
            "admin_state_up": true,
            "allowed_address_pairs": [],
            "binding:host_id": "devstack",
            "binding:profile": {},
            "binding:vif_details": {
                "ovs_hybrid_plug": true,
                "port_filter": true
            },
            "binding:vif_type": "ovs",
            "binding:vnic_type": "normal",
            "created_at": "2016-03-08T20:19:41",
            "data_plane_status": null,
            "description": "",
            "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824",
            "device_owner": "network:router_interface",
            "dns_assignment": {
                "hostname": "myport2",
                "ip_address": "10.0.0.1",
                "fqdn": "myport2.my-domain.org"
            },
            "dns_domain": "my-domain.org.",
            "dns_name": "myport2",
            "extra_dhcp_opts": [],
            "fixed_ips": [
                {
                    "ip_address": "10.0.0.1",
                    "subnet_id": "288bf4a1-51ba-43b6-9d0a-520e9005db17"
                }
            ],
            "id": "f71a6703-d6de-4be1-a91a-a570ede1d159",
            "ip_allocation": "immediate",
            "mac_address": "fa:16:3e:bb:3c:e4",
            "name": "",
            "network_id": "f27aa545-cbdd-4907-b0c6-c9e8b039dcc2",
            "port_security_enabled": true,
            "project_id": "d397de8a63f341818f198abb0966f6f3",
            "revision_number": 2,
            "security_groups": [],
            "status": "ACTIVE",
            "tenant_id": "d397de8a63f341818f198abb0966f6f3",
            "updated_at": "2016-03-08T20:19:41",
            "qos_policy_id": null,
            "tags": ["tag1,tag2"],
            "tenant_id": "d397de8a63f341818f198abb0966f6f3",
            "uplink_status_propagation": false
        }
    ]
}
```

Create port POST /v2.0/ports

Creates a port on a network.

To define the network in which to create the port, specify the `network_id` attribute in the request body.

Normal response codes: 201

Error response codes: 400, 401, 403, 404

Request¶

Name	In	Type	Description
port	body	object	A `port` object.
admin_state_up (Optional)	body	boolean	The administrative state of the resource, which is up (`true`) or down (`false`). Default is `true` .
allowed_address_pairs (Optional)	body	array	A set of zero or more allowed address pair objects each where address pair object contains an `ip_address` and `mac_address` . While the `ip_address` is required, the `mac_address` will be taken from the port if not specified. The value of `ip_address` can be an IP Address or a CIDR (if supported by the underlying extension plugin). A server connected to the port can send a packet with source address which matches one of the specified allowed address pairs.
binding:host_id (Optional)	body	string	The ID of the host where the port resides. The default is an empty string.
binding:profile (Optional)	body	object	A dictionary that enables the application running on the specific host to pass and receive vif port information specific to the networking back-end. The networking API does not define a specific format of this field. The default is an empty dictionary.
binding:vnic_type (Optional)	body	string	The type of vNIC which this port should be attached to. This is used to determine which mechanism driver(s) to be used to bind the port. The valid values are `normal` , `macvtap` , `direct` , `baremetal` , `direct-physical` and `virtio-forwarder` . What type of vNIC is actually available depends on deployments. The default is `normal` .
description (Optional)	body	string	A human-readable description for the resource. Default is an empty string.
device_id (Optional)	body	string	The ID of the device that uses this port. For example, a server instance or a logical router.
device_owner (Optional)	body	string	The entity type that uses this port. For example, `compute:nova` (server instance), `network:dhcp` (DHCP agent) or `network:router_interface` (router interface).
dns_domain (Optional)	body	string	A valid DNS domain.
dns_name (Optional)	body	string	A valid DNS name.
extra_dhcp_opts (Optional)	body	array	A set of zero or more extra DHCP option pairs. An option pair consists of an option value and name.
fixed_ips (Optional)	body	array	The IP addresses for the port. If you would like to assign multiple IP addresses for the port, specify multiple entries in this field. Each entry consists of IP address (`ip_address`) and the subnet ID from which the IP address is assigned (`subnet_id`).If you specify both a subnet ID and an IP address, OpenStack Networking tries to allocate the IP address on that subnet to the port.If you specify only a subnet ID, OpenStack Networking allocates an available IP from that subnet to the port.If you specify only an IP address, OpenStack Networking tries to allocate the IP address if the address is a valid IP for any of the subnets on the specified network.
mac_address (Optional)	body	string	The MAC address of the port. If unspecified, a MAC address is automatically generated.
name (Optional)	body	string	Human-readable name of the resource. Default is an empty string.
network_id	body	string	The ID of the attached network.
port_security_enabled (Optional)	body	boolean	The port security status. A valid value is enabled (`true`) or disabled (`false`). If port security is enabled for the port, security group rules and anti-spoofing rules are applied to the traffic on the port. If disabled, no such rules are applied.
project_id (Optional)	body	string	The ID of the project that owns the resource. Only administrative and users with advsvc role can specify a project ID other than their own. You cannot change this value through authorization policies.
qos_policy_id (Optional)	body	string	QoS policy associated with the port.
security_groups (Optional)	body	array	The IDs of security groups applied to the port.
tags	body	array	The list of tags on the resource.
tenant_id (Optional)	body	string	The ID of the project that owns the resource. Only administrative and users with advsvc role can specify a project ID other than their own. You cannot change this value through authorization policies.
uplink_status_propagation (Optional)	body	boolean	The uplink status propagation of the port. Valid values are enabled (`true`) and disabled (`false`).
mac_learning_enabled (Optional)	body	boolean	A boolean value that indicates if MAC Learning is enabled on the associated port.

Request Example¶

```
{
    "port": {
        "admin_state_up": true,
        "dns_domain": "my-domain.org.",
        "dns_name": "myport",
        "name": "private-port",
        "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
        "port_security_enabled": true,
        "allowed_address_pairs": [
            {
                "ip_address": "12.12.11.12",
                "mac_address": "fa:14:2a:b3:cb:f0"
            }
        ],
        "uplink_status_propagation": false
    }
}
```

Request Example (admin user)¶

```
{
    "port": {
        "binding:host_id": "4df8d9ff-6f6f-438f-90a1-ef660d4586ad",
        "binding:profile": {
            "local_link_information": [
                {
                    "port_id": "Ethernet3/1",
                    "switch_id": "0a:1b:2c:3d:4e:5f",
                    "switch_info": "switch1"
                }
            ]
        },
        "binding:vnic_type": "baremetal",
        "device_id": "d90a13da-be41-461f-9f99-1dbcf438fdf2",
        "device_owner": "baremetal:none",
        "dns_domain": "my-domain.org.",
        "dns_name": "myport",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
        "uplink_status_propagation": false
    }
}
```

Response Parameters¶

Name	In	Type	Description
port	body	object	A `port` object.
admin_state_up	body	boolean	The administrative state of the resource, which is up (`true`) or down (`false`).
allowed_address_pairs	body	array	A set of zero or more allowed address pair objects each where address pair object contains an `ip_address` and `mac_address` . While the `ip_address` is required, the `mac_address` will be taken from the port if not specified. The value of `ip_address` can be an IP Address or a CIDR (if supported by the underlying extension plugin). A server connected to the port can send a packet with source address which matches one of the specified allowed address pairs.
binding:host_id	body	string	The ID of the host where the port resides.
binding:profile	body	object	A dictionary that enables the application running on the specific host to pass and receive vif port information specific to the networking back-end. The networking API does not define a specific format of this field.
binding:vif_details	body	object	A dictionary which contains additional information on the port. Currently the following fields are defined: `port_filter` and `ovs_hybrid_plug` . `port_filter` is a boolean indicating the networking service provides port filtering features such as security group and/or anti MAC/IP spoofing. `ovs_hybrid_plug` is a boolean used to inform an API consumer like nova that the hybrid plugging strategy for OVS should be used.
binding:vif_type	body	string	The type of which mechanism is used for the port. An API consumer like nova can use this to determine an appropriate way to attach a device (for example an interface of a virtual server) to the port. Available values currently defined includes `ovs` , `bridge` , `macvtap` , `hw_veb` , `hostdev_physical` , `vhostuser` , `distributed` and `other` . There are also special values: `unbound` and `binding_failed` . `unbound` means the port is not bound to a networking back-end. `binding_failed` means an error that the port failed to be bound to a networking back-end.

binding:vnic_type	body	string	The type of vNIC which this port should be attached to. This is used to determine which mechanism driver(s) to be used to bind the port. The valid values are `normal`, `macvtap`, `direct`, `baremetal`, `direct-physical` and `virtio-forwarder`. What type of vNIC is actually available depends on deployments.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
data_plane_status	body	string	Status of the underlying data plane of a port.
description	body	string	A human-readable description for the resource.
device_id	body	string	The ID of the device that uses this port. For example, a server instance or a logical router.
device_owner	body	string	The entity type that uses this port. For example, `compute:nova` (server instance), `network:dhcp` (DHCP agent) or `network:router_interface` (router interface).
dns_assignment	body	object	Data assigned to a port by the Networking internal DNS including the `hostname`, `ip_address` and `fqdn`.
dns_domain	body	string	A valid DNS domain.
dns_name	body	string	A valid DNS name.
extra_dhcp_opts	body	array	A set of zero or more extra DHCP option pairs. An option pair consists of an option value and name.
fixed_ips	body	array	The IP addresses for the port. If the port has multiple IP addresses, this field has multiple entries. Each entry consists of IP address (`ip_address`) and the subnet ID from which the IP address is assigned (`subnet_id`).
id	body	string	The ID of the resource.
ip_allocation	body	string	Indicates when ports use either `deferred`, `immediate` or no IP allocation (`none`).
mac_address	body	string	The MAC address of the port.
name	body	string	Human-readable name of the resource.
network_id	body	string	The ID of the attached network.
port_security_enabled	body	boolean	The port security status. A valid value is enabled (`true`) or disabled (`false`). If port security is enabled for the port, security group rules and anti-spoofing rules are applied to the traffic on the port. If disabled, no such rules are applied.
project_id	body	string	The ID of the project.
revision_number	body	integer	The revision number of the resource.
qos_policy_id	body	string	The ID of the QoS policy associated with the port.
resource_request (Optional)	body	object	Expose Placement resources (i.e.: `minimum-bandwidth`) and traits (i.e.: `vnic-type`, `physnet`) requested by a port to Nova and Placement. A `resource_request` object contains a `required` key for the traits (generated from the `vnic_type` and the `physnet`) required by the port, and a `resources` key for `ingress` and `egress` `minimum-bandwidth` need for the port.
security_groups	body	array	The IDs of security groups applied to the port.
status	body	string	The port status. Values are `ACTIVE`, `DOWN`, `BUILD` and `ERROR`.
tags	body	array	The list of tags on the resource.
tenant_id	body	string	The ID of the project.
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
uplink_status_propagation	body	boolean	The uplink status propagation of the port. Valid values are enabled (`true`) and disabled (`false`).
mac_learning_enabled (Optional)	body	boolean	A boolean value that indicates if MAC Learning is enabled on the associated port.

Response Example¶

```
{
    "port": {
        "admin_state_up": true,
        "allowed_address_pairs": [
            {
                "ip_address": "12.12.11.12",
                "mac_address": "fa:14:2a:b3:cb:f0"
            }
        ],
        "created_at": "2016-03-08T20:19:41",
        "data_plane_status": null,
        "description": "",
        "device_id": "",
        "device_owner": "",
        "dns_assignment": {
            "hostname": "myport",
            "ip_address": "10.0.0.2",
            "fqdn": "myport.my-domain.org"
        },
        "dns_domain": "my-domain.org.",
        "dns_name": "myport",
        "extra_dhcp_opts": [
            {
                "opt_value": "pxelinux.0",
                "ip_version": 4,
                "opt_name": "bootfile-name"
            }
        ],
        "fixed_ips": [
            {
                "ip_address": "10.0.0.2",
                "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2"
            }
        ],
        "id": "65c0ee9f-d634-4522-8954-51021b570b0d",
        "ip_allocation": "immediate",
        "mac_address": "fa:16:3e:c9:cb:f0",
        "name": "private-port",
        "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
        "port_security_enabled": true,
        "project_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
        "revision_number": 1,
        "security_groups": [
            "f0ac4394-7e4a-4409-9701-ba8be283dbc3"
        ],
        "status": "DOWN",
        "tags": ["tag1,tag2"],
        "tenant_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
        "updated_at": "2016-03-08T20:19:41",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
        "uplink_status_propagation": false
    }
}
```

Response Example (admin user)¶

```
{
    "port": {
        "admin_state_up": true,
        "allowed_address_pairs": [
            {
                "ip_address": "12.12.11.12",
                "mac_address": "fa:14:2a:b3:cb:f0"
            }
        ],
        "binding:host_id": "4df8d9ff-6f6f-438f-90a1-ef660d4586ad",
        "binding:profile": {
            "local_link_information": [
                {
                    "port_id": "Ethernet3/1",
                    "switch_id": "0a:1b:2c:3d:4e:5f",
                    "switch_info": "switch1"
                }
            ]
        },
        "binding:vif_details": {},
        "binding:vif_type": "unbound",
        "binding:vnic_type": "other",
        "created_at": "2016-03-08T20:19:41",
        "data_plane_status": null,
        "description": "",
        "device_id": "d90a13da-be41-461f-9f99-1dbcf438fdf2",
        "device_owner": "baremetal:none",
        "dns_assignment": {
            "hostname": "myport",
            "ip_address": "10.0.0.2",
            "fqdn": "myport.my-domain.org"
        },
        "dns_domain": "my-domain.org.",
        "dns_name": "myport",
        "extra_dhcp_opts": [
            {
                "opt_value": "pxelinux.0",
                "ip_version": 4,
                "opt_name": "bootfile-name"
            }
        ],
        "fixed_ips": [
            {
                "ip_address": "10.0.0.2",
                "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2"
            }
        ],
        "id": "65c0ee9f-d634-4522-8954-51021b570b0d",
        "ip_allocation": "immediate",
        "mac_address": "fa:16:3e:c9:cb:f0",
        "name": "private-port",
        "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
        "project_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
        "revision_number": 1,
        "security_groups": [
            "f0ac4394-7e4a-4409-9701-ba8be283dbc3"
        ],
        "status": "DOWN",
        "tags": ["tag1,tag2"],
        "tenant_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
        "updated_at": "2016-03-08T20:19:41",
        "qos_policy_id": "29d5e02e-d5ab-4929-bee4-4a9fc12e22ae",
        "port_security_enabled": true,
        "resource_request": {
            "required": ["CUSTOM_PHYSNET_PUBLIC", "CUSTOM_VNIC_TYPE_NORMAL"],
            "resources": {"NET_BW_EGR_KILOBIT_PER_SEC": 1000}
        },
        "uplink_status_propagation": false
    }
}
```

Layer 3 Networking¶

Address scopes¶

Lists, creates, shows details for, updates, and deletes address scopes.

GET

/v2.0/address-scopes/{address_scope_id}

Show address scope

detail

PUT

/v2.0/address-scopes/{address_scope_id}

Update an address scope

detail

DELETE

/v2.0/address-scopes/{address_scope_id}

Delete an address scope

detail

GET

/v2.0/address-scopes

List address scopes

detail

POST

/v2.0/address-scopes

Create address scope

detail

Floating IPs (floatingips)¶

DNS integration¶

The `dns-integration` extension adds the `dns_name` and `dns_domain` attributes to floating IPs allowing them to be specified at creation time. The data in these attributes will be published in an external DNS service when Neutron is configured to integrate with such a service.

Floating IP port details¶

The `fip-port-details` extension adds the `port_details` attribute to floating IPs. The value of this attribute contains information of the associated port.

Floating IP port forwardings¶

The `expose-port-forwarding-in-fip` extension adds the `port_forwardings` attribute to floating IPs. The value of this attribute contains the information of associated port forwarding resources.

Resource timestamps¶

The `standard-attr-timestamp` extension adds the `created_at` and `updated_at` attributes to all resources that have standard attributes.

Tag extension¶

The `standard-attr-tag` adds Tag support for resources with standard attributes by adding the `tags` attribute allowing consumers to associate tags with resources.

List floating IPs GET /v2.0/floatingips

Lists floating IPs visible to the user.

Default policy settings return only the floating IPs owned by the user's project, unless the user has admin role.

This example request lists floating IPs in JSON format:

```
GET /v2.0/floatingips
Accept: application/json
```

Use the `fields` query parameter to control which fields are returned in the response body. Additionally, you can filter results by using query string parameters. For information, see [Filtering and Column Selection](#).

Normal response codes: 200

Error response codes: 401

Request¶

Name	In	Type	Description
id (Optional)	query	string	Filter the list result by the ID of the resource.
router_id (Optional)	query	string	Filter the floating IP list result by the ID of the router for the floating IP.
status (Optional)	query	string	Filter the floating IP list result by the status of the floating IP. Values are `ACTIVE`, `DOWN` and `ERROR`.
tenant_id (Optional)	query	string	Filter the list result by the ID of the project that owns the resource.
project_id (Optional)	query	string	Filter the list result by the ID of the project that owns the resource.
revision_number (Optional)	query	integer	Filter the list result by the revision number of the resource.
description (Optional)	query	string	Filter the list result by the human-readable description of the resource.
floating_network_id (Optional)	query	string	Filter the floating IP list result by the ID of the network associated with the floating IP.
fixed_ip_address (Optional)	query	string	Filter the floating IP list result by the fixed IP address that is associated with the floating IP address.
floating_ip_address (Optional)	query	string	Filter the floating IP list result by the floating IP address.
port_id (Optional)	query	string	Filter the floating IP list result by the ID of a port associated with the floating IP.
sort_dir (Optional)	query	string	Sort direction. A valid value is `asc` (ascending) or `desc` (descending). You can specify multiple pairs of sort key and sort direction query parameters.
sort_key (Optional)	query	string	Sorts by a floatingip attribute. You can specify multiple pairs of sort key and sort direction query parameters. The sort keys are limited to: fixed_ip_address``floating_ip_address``floating_network_id``id``router_id``status``tenant_id``project_id
tags (Optional)	query	string	A list of tags to filter the list result by. Resources that match all tags in this list will be returned. Tags in query must be separated by comma.
tags-any (Optional)	query	string	A list of tags to filter the list result by. Resources that match any tag in this list will be returned. Tags in query must be separated by comma.
not-tags (Optional)	query	string	A list of tags to filter the list result by. Resources that match all tags in this list will be excluded. Tags in query must be separated by comma.
not-tags-any (Optional)	query	string	A list of tags to filter the list result by. Resources that match any tag in this list will be excluded. Tags in query must be separated by comma.
fields (Optional)	query	string	The fields that you want the server to return. If no `fields` query parameter is specified, the networking API returns all attributes allowed by the policy settings. By using `fields` parameter, the API returns only the requested set of attributes. `fields` parameter can be specified multiple times. For example, if you specify `fields=id&fields=name` in the request URL, only `id` and `name` attributes will be returned.

Response Parameters¶

Name	In	Type	Description
floatingips	body	array	A list of `floatingip` objects.
id	body	string	The ID of the floating IP address.
router_id	body	string	The ID of the router for the floating IP.
status	body	string	The status of the floating IP. Values are `ACTIVE`, `DOWN` and `ERROR`.
tenant_id	body	string	The ID of the project.
project_id	body	string	The ID of the project.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
revision_number	body	integer	The revision number of the resource.
description	body	string	A human-readable description for the resource.
dns_domain	body	string	A valid DNS domain.
dns_name	body	string	A valid DNS name.
port_details	body	string	The information of the port that this floating IP associates with. In particular, if the floating IP is associated with a port, this field contains some attributes of the associated port, including `name`, `network_id`, `mac_address`, `admin_state_up`, `status`, `device_id` and `device_owner`. If the floating IP is not associated with a port, this field is `null`.
floating_network_id	body	string	The ID of the network associated with the floating IP.
fixed_ip_address	body	string	The fixed IP address that is associated with the floating IP address.
floating_ip_address	body	string	The floating IP address.
port_id	body	string	The ID of a port associated with the floating IP.
tags	body	array	The list of tags on the resource.
port_forwardings	body	array	The associated port forwarding resources for the floating IP. If the floating IP has multiple port forwarding resources, this field has multiple entries. Each entry consists of network IP protocol (`protocol`), the fixed IP address of internal neutron port (`internal_ip_address`), the TCP or UDP port used by internal neutron port (`internal_port`) and the TCP or UDP port used by floating IP (`external_port`).

Response Example¶

```
{
    "floatingips": [
        {
            "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
            "description": "for test",
            "dns_domain": "my-domain.org.",
            "dns_name": "myfip",
            "created_at": "2016-12-21T10:55:50Z",
            "updated_at": "2016-12-21T10:55:53Z",
            "revision_number": 1,
            "project_id": "4969c491a3c74ee4af974e6d800c62de",
            "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
            "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
            "fixed_ip_address": "10.0.0.3",
            "floating_ip_address": "172.24.4.228",
            "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
            "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
            "status": "ACTIVE",
            "port_details": {
                "status": "ACTIVE",
                "name": "",
                "admin_state_up": true,
                "network_id": "02dd8479-ef26-4398-a102-d19d0a7b3a1f",
                "device_owner": "compute:nova",
                "mac_address": "fa:16:3e:b1:3b:30",
                "device_id": "8e3941b4-a6e9-499f-a1ac-2a4662025cba"
            },
            "tags": ["tag1,tag2"],
            "port_forwardings": []
        },
        {
            "router_id": null,
            "description": "for test",
            "dns_domain": "my-domain.org.",
            "dns_name": "myfip2",
            "created_at": "2016-12-21T11:55:50Z",
            "updated_at": "2016-12-21T11:55:53Z",
            "revision_number": 2,
            "project_id": "4969c491a3c74ee4af974e6d800c62de",
            "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
            "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
            "fixed_ip_address": null,
            "floating_ip_address": "172.24.4.227",
            "port_id": null,
            "id": "61cea855-49cb-4846-997d-801b70c71bdd",
            "status": "DOWN",
            "port_details": null,
            "tags": ["tag1,tag2"],
            "port_forwardings": []
        },
        {
            "router_id": "0303bf18-2c52-479c-bd68-e0ad712a1639",
            "description": "for test with port forwarding",
            "dns_domain": "my-domain.org.",
            "dns_name": "myfip3",
            "created_at": "2018-06-15T02:12:48Z",
            "updated_at": "2018-06-15T02:12:57Z",
            "revision_number": 1,
            "project_id": "4969c491a3c74ee4af974e6d800c62de",
            "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
            "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
            "fixed_ip_address": null,
            "floating_ip_address": "172.24.4.42",
            "port_id": null,
            "id": "898b198e-49f7-47d6-a7e1-53f626a548e6",
            "status": "ACTIVE",
            "tags": [],
            "port_forwardings": [
                {
                    "protocol": "tcp",
                    "internal_ip_address": "10.0.0.19",
                    "internal_port": 25,
                    "external_port": 2225
                },
                {
                    "protocol": "tcp",
                    "internal_ip_address": "10.0.0.18",
                    "internal_port": 16666,
                    "external_port": 8786
                }
            ]
        }
    ]
}
```

Create floating IP POST /v2.0/floatingips

Creates a floating IP, and, if you specify port information, associates the floating IP with an internal port.

To associate the floating IP with an internal port, specify the port ID attribute in the request body. If you do not specify a port ID in the request, you can issue a PUT request instead of a POST request.

Default policy settings enable only administrative users to set floating IP addresses and some non-administrative users might require a floating IP address. If you do not specify a floating IP address in the request, the operation automatically allocates one.

By default, this operation associates the floating IP address with a single fixed IP address that is configured on an OpenStack Networking port. If a port has multiple IP addresses, you must specify the `fixed_ip_address` attribute in the request body to associate a fixed IP address with the floating IP address.

You can create floating IPs on only external networks. When you create a floating IP, you must specify the ID of the network on which you want to create the floating IP. Alternatively, you can create a floating IP on a subnet in the external network, based on the costs and quality of that subnet.

You must configure an IP address with the internal OpenStack Networking port that is associated with the floating IP address.

The operation returns the `Bad Request (400)` response code for one of reasons:

> - The network is not external, such as `router:external=False`.
> - The internal OpenStack Networking port is not associated with the floating IP address.
> - The requested floating IP address does not fall in the subnet range for the external network.
> - The fixed IP address is not valid.

If the port ID is not valid, this operation returns `404` response code.

The operation returns the `Conflict (409)` response code for one of reasons:

> - The requested floating IP address is already in use.
> - The internal OpenStack Networking port and fixed IP address are already associated with another floating IP.

Normal response codes: 201

Error response codes: 400, 401, 404, 409

Request¶

Name	In	Type	Description
floatingip	body	object	A `floatingip` object. When you associate a floating IP address with a VM, the instance has the same public IP address each time that it boots, basically to maintain a consistent IP address for maintaining DNS assignment.
tenant_id	body	string	The ID of the project.
project_id	body	string	The ID of the project.
floating_network_id	body	string	The ID of the network associated with the floating IP.
fixed_ip_address (Optional)	body	string	The fixed IP address that is associated with the floating IP. If an internal port has multiple associated IP addresses, the service chooses the first IP address unless you explicitly define a fixed IP address in the `fixed_ip_address` parameter.
floating_ip_address (Optional)	body	string	The floating IP address.
port_id (Optional)	body	string	The ID of a port associated with the floating IP. To associate the floating IP with a fixed IP at creation time, you must specify the identifier of the internal port.
subnet_id (Optional)	body	string	The subnet ID on which you want to create the floating IP.
description (Optional)	body	string	A human-readable description for the resource. Default is an empty string.
dns_domain (Optional)	body	string	A valid DNS domain.
dns_name (Optional)	body	string	A valid DNS name.

Request Example¶

```
{
    "floatingip": {
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
        "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
        "subnet_id": "278d9507-36e7-403c-bb80-1d7093318fe6",
        "fixed_ip_address": "10.0.0.3",
        "floating_ip_address": "172.24.4.228",
        "description": "floating ip for testing",
        "dns_domain": "my-domain.org.",
        "dns_name": "myfip"
    }
}
```

Response Parameters¶

Name	In	Type	Description
floatingip	body	object	A `floatingip` object. When you associate a floating IP address with a VM, the instance has the same public IP address each time that it boots, basically to maintain a consistent IP address for maintaining DNS assignment.
router_id	body	string	The ID of the router for the floating IP.
status	body	string	The status of the floating IP. Values are `ACTIVE`, `DOWN` and `ERROR`.
description	body	string	A human-readable description for the resource.
dns_domain	body	string	A valid DNS domain.
dns_name	body	string	A valid DNS name.
port_details	body	string	The information of the port that this floating IP associates with. In particular, if the floating IP is associated with a port, this field contains some attributes of the associated port, including `name`, `network_id`, `mac_address`, `admin_state_up`, `status`, `device_id` and `device_owner`. If the floating IP is not associated with a port, this field is `null`.
tenant_id	body	string	The ID of the project.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
revision_number	body	integer	The revision number of the resource.
project_id	body	string	The ID of the project.
floating_network_id	body	string	The ID of the network associated with the floating IP.
fixed_ip_address	body	string	The fixed IP address that is associated with the floating IP address.
floating_ip_address	body	string	The floating IP address.
port_id	body	string	The ID of a port associated with the floating IP.
id	body	string	The ID of the floating IP address.
tags	body	array	The list of tags on the resource.
port_forwardings	body	array	The associated port forwarding resources for the floating IP. If the floating IP has multiple port forwarding resources, this field has multiple entries. Each entry consists of network IP protocol (`protocol`), the fixed IP address of internal neutron port (`internal_ip_address`), the TCP or UDP port used by internal neutron port (`internal_port`) and the TCP or UDP port used by floating IP (`external_port`).

Response Example¶

```
{
    "floatingip": {
        "fixed_ip_address": "10.0.0.3",
        "floating_ip_address": "172.24.4.228",
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
        "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
        "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
        "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
        "status": "ACTIVE",
        "project_id": "4969c491a3c74ee4af974e6d800c62de",
        "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
        "description": "floating ip for testing",
        "dns_domain": "my-domain.org.",
        "dns_name": "myfip",
        "created_at": "2016-12-21T01:36:04Z",
        "updated_at": "2016-12-21T01:36:04Z",
        "revision_number": 1,
        "port_details": {
            "status": "ACTIVE",
            "name": "",
            "admin_state_up": true,
            "network_id": "02dd8479-ef26-4398-a102-d19d0a7b3a1f",
            "device_owner": "compute:nova",
            "mac_address": "fa:16:3e:b1:3b:30",
            "device_id": "8e3941b4-a6e9-499f-a1ac-2a4662025cba"
        },
        "tags": ["tag1,tag2"],
        "port_forwardings": []
    }
}
```

Show floating IP details GET /v2.0/floatingips/{floatingip_id}

Shows details for a floating IP.

Use the `fields` query parameter to control which fields are returned in the response body. For information, see Filtering and Column Selection.

This example request shows details for a floating IP in JSON format. This example also filters the result by the `fixed_ip_address` and `floating_ip_address` fields.

```
GET /v2.0/floatingips/{floatingip_id}?fields=fixed_ip_address
&
fields=floating_ip_address
Accept: application/json
```

Normal response codes: 200

Error response codes: 401, 403, 404

Request¶

Name	In	Type	Description
floatingip_id	path	string	The ID of the floating IP address.

Response Parameters¶

Name	In	Type	Description
floatingip	body	object	A `floatingip` object. When you associate a floating IP address with a VM, the instance has the same public IP address each time that it boots, basically to maintain a consistent IP address for maintaining DNS assignment.
router_id	body	string	The ID of the router for the floating IP.
status	body	string	The status of the floating IP. Values are `ACTIVE`, `DOWN` and `ERROR`.
description	body	string	A human-readable description for the resource.
dns_domain	body	string	A valid DNS domain.
dns_name	body	string	A valid DNS name.
port_details	body	string	The information of the port that this floating IP associates with. In particular, if the floating IP is associated with a port, this field contains some attributes of the associated port, including `name`, `network_id`, `mac_address`, `admin_state_up`, `status`, `device_id` and `device_owner`. If the floating IP is not associated with a port, this field is `null`.
tenant_id	body	string	The ID of the project.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
revision_number	body	integer	The revision number of the resource.
project_id	body	string	The ID of the project.
floating_network_id	body	string	The ID of the network associated with the floating IP.
fixed_ip_address	body	string	The fixed IP address that is associated with the floating IP address.
floating_ip_address	body	string	The floating IP address.
port_id	body	string	The ID of a port associated with the floating IP.
id	body	string	The ID of the floating IP address.
tags	body	array	The list of tags on the resource.
port_forwardings	body	array	The associated port forwarding resources for the floating IP. If the floating IP has multiple port forwarding resources, this field has multiple entries. Each entry consists of network IP protocol (`protocol`), the fixed IP address of internal neutron port (`internal_ip_address`), the TCP or UDP port used by internal neutron port (`internal_port`) and the TCP or UDP port used by floating IP (`external_port`).

Response Example¶

```
{
    "floatingip": {
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
        "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
        "fixed_ip_address": "10.0.0.3",
        "floating_ip_address": "172.24.4.228",
        "project_id": "4969c491a3c74ee4af974e6d800c62de",
        "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
        "status": "ACTIVE",
        "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
        "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
        "description": "floating ip for testing",
        "dns_domain": "my-domain.org.",
        "dns_name": "myfip",
        "created_at": "2016-12-21T01:36:04Z",
        "updated_at": "2016-12-21T01:36:04Z",
        "revision_number": 1,
        "port_details": {
            "status": "ACTIVE",
            "name": "",
            "admin_state_up": true,
            "network_id": "02dd8479-ef26-4398-a102-d19d0a7b3a1f",
            "device_owner": "compute:nova",
            "mac_address": "fa:16:3e:b1:3b:30",
            "device_id": "8e3941b4-a6e9-499f-a1ac-2a4662025cba"
        },
        "tags": ["tag1,tag2"],
        "port_forwardings": []

    }
}
```

Update floating IP PUT /v2.0/floatingips/{floatingip_id}

Updates a floating IP and its association with an internal port.

The association process is the same as the process for the create floating IP operation.

To disassociate a floating IP from a port, set the `port_id` attribute to null or omit it from the request body.

This example updates a floating IP:

```
PUT /v2.0/floatingips/{floatingip_id} Accept: application/json
```

Depending on the request body that you submit, this request associates a port with or disassociates a port from a floating IP.

Normal response codes: 200

Error response codes: 400, 401, 404, 409, 412

Request¶

Name	In	Type	Description
floatingip	body	object	A `floatingip` object. When you associate a floating IP address with a VM, the instance has the same public IP address each time that it boots, basically to maintain a consistent IP address for maintaining DNS assignment.
floatingip_id	path	string	The ID of the floating IP address.
port_id	body	string	The ID of a port associated with the floating IP. To associate the floating IP with a fixed IP, you must specify the ID of the internal port. To disassociate the floating IP, `null` should be specified.
fixed_ip_address (Optional)	body	string	The fixed IP address that is associated with the floating IP. If an internal port has multiple associated IP addresses, the service chooses the first IP address unless you explicitly define a fixed IP address in the `fixed_ip_address` parameter.
description (Optional)	body	string	A human-readable description for the resource. Default is an empty string.

Request Example¶

```
{
    "floatingip": {
        "port_id": "fc861431-0e6c-4842-a0ed-e2363f9bc3a8"
    }
}
```

Request Example (disassociate)¶

```
{
    "floatingip": {
        "port_id": null
    }
}
```

Response Parameters¶

Name	In	Type	Description
floatingip	body	object	A `floatingip` object. When you associate a floating IP address with a VM, the instance has the same public IP address each time that it boots, basically to maintain a consistent IP address for maintaining DNS assignment.
router_id	body	string	The ID of the router for the floating IP.
status	body	string	The status of the floating IP. Values are `ACTIVE`, `DOWN` and `ERROR`.
tenant_id	body	string	The ID of the project.
project_id	body	string	The ID of the project.
floating_network_id	body	string	The ID of the network associated with the floating IP.
fixed_ip_address	body	string	The fixed IP address that is associated with the floating IP address.
floating_ip_address	body	string	The floating IP address.
port_id	body	string	The ID of a port associated with the floating IP.
id	body	string	The ID of the floating IP address.
created_at	body	string	Time at which the resource has been created (in UTC ISO8601 format).
updated_at	body	string	Time at which the resource has been updated (in UTC ISO8601 format).
revision_number	body	integer	The revision number of the resource.
description	body	string	A human-readable description for the resource.
dns_domain	body	string	A valid DNS domain.
dns_name	body	string	A valid DNS name.
port_details	body	string	The information of the port that this floating IP associates with. In particular, if the floating IP is associated with a port, this field contains some attributes of the associated port, including `name`, `network_id`, `mac_address`, `admin_state_up`, `status`, `device_id` and `device_owner`. If the floating IP is not associated with a port, this field is `null`.
tags	body	array	The list of tags on the resource.
port_forwardings	body	array	The associated port forwarding resources for the floating IP. If the floating IP has multiple port forwarding resources, this field has multiple entries. Each entry consists of network IP protocol (`protocol`), the fixed IP address of internal neutron port (`internal_ip_address`), the TCP or UDP port used by internal neutron port (`internal_port`) and the TCP or UDP port used by floating IP (`external_port`).

Response Example¶

```
{
    "floatingip": {
        "created_at": "2016-12-21T10:55:50Z",
        "description": "floating ip for testing",
        "dns_domain": "my-domain.org.",
        "dns_name": "myfip",
        "fixed_ip_address": "10.0.0.4",
        "floating_ip_address": "172.24.4.228",
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
        "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
        "port_id": "fc861431-0e6c-4842-a0ed-e2363f9bc3a8",
        "project_id": "4969c491a3c74ee4af974e6d800c62de",
        "revision_number": 3,
        "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
        "status": "ACTIVE",
        "tags": ["tag1,tag2"],
        "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
        "updated_at": "2016-12-22T03:13:49Z",
        "port_details": {
            "status": "ACTIVE",
            "name": "",
            "admin_state_up": true,
            "network_id": "02dd8479-ef26-4398-a102-d19d0a7b3a1f",
            "device_owner": "compute:nova",
            "mac_address": "fa:16:3e:b1:3b:30",
            "device_id": "8e3941b4-a6e9-499f-a1ac-2a4662025cba"
        },
        "port_forwardings": []
    }
}
```

Response Example (disassociate)¶

```
{
    "floatingip": {
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
        "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
        "fixed_ip_address": null,
        "floating_ip_address": "172.24.4.228",
        "project_id": "4969c491a3c74ee4af974e6d800c62de",
        "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
        "status": "ACTIVE",
        "port_id": null,
        "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
        "description": "for test",
        "created_at": "2016-12-21T10:55:50Z",
        "updated_at": "2016-12-22T03:13:49Z",
        "revision_number": 3,
        "port_details": null,
        "tags": ["tag1,tag2"],
        "port_forwardings": []
    }
}
```

Delete floating IP DELETE /v2.0/floatingips/{floatingip_id}

Deletes a floating IP and, if present, its associated port.

This example deletes a floating IP:

```
DELETE /v2.0/floatingips/{floatingip_id} Accept: application/json
```

Normal response codes: 204

Error response codes: 401, 404, 412

Request¶

Name	In	Type	Description
floatingip_id	path	string	The ID of the floating IP address.

Response¶

There is no body content for the response of a successful DELETE request.