



**Elastic Cloud Server**

# **Developer Guide**

**Date**      **2018-11-08**

---

# Contents

---

<b>1 Overview.....</b>	<b>1</b>
<b>2 Notes.....</b>	<b>3</b>
<b>3 Request Format.....</b>	<b>4</b>
3.1 Versions.....	5
3.2 Microversions.....	5
3.3 Request Example.....	9
<b>4 Creating an ECS.....</b>	<b>14</b>
<b>5 Modifying ECS Specifications.....</b>	<b>21</b>
<b>6 Attaching a Disk to an ECS.....</b>	<b>23</b>
<b>7 Binding a NIC to an ECS.....</b>	<b>26</b>
<b>A Change History.....</b>	<b>31</b>

# 1 Overview

---

This document describes how to call the APIs of Elastic Cloud Server (ECS) to use ECS functions. The concepts related to ECS help you quickly learn this service.

## ECS

An ECS is a cloud server that allows on-demand allocation and elastic scaling to create an efficient, reliable, and secure computing environment. This ensures stable and uninterrupted operation of services.

## Basic Concepts

- **Region**  
A region is a geographic area where resources used by ECSs are located.
- **Availability zone (AZ)**  
An AZ is a physical location where power and networks are physically isolated within a region. Each AZ provides cost-effective and low-latency network connections that are unaffected by faults that may occur in other AZs. Each region contains one or more AZs. AZs are physically isolated but interconnected through an internal network.
- **Project**  
A project groups and isolates OpenStack resources, such as computing, storage, and network resources. A project can be either a department or a project team.  
A tenant can create multiple projects.
- **Flavor**  
Specifies hardware resources required for running an ECS, including the vCPUs, memory, and storage capacities.
- **Elastic Volume Service (EVS)**  
Provides persistent block storage for computing services, such as ECS and Bare Metal Server (BMS). With advanced data redundancy and cache acceleration capabilities, EVS offers high availability and durability with a low latency. Users can format an EVS disk, create a file system on it, and store data persistently.
- **Image**  
An image is an ECS template that contains an OS and may also contain application software (such as database software) and software configuration.

Images can be public or private. Public images are provided by the system by default, and private images are manually created by users. Users can use any type of image to create an ECS. They can also create a private image using an existing ECS or external image. This provides users with a simple way to create ECSs that comply with their service requirements.

- Virtual Private Cloud (VPC)

VPC allows users to create private, isolated virtual networks. Users can define IP address segments, subnets, and security groups, assign EIPs, and allocate bandwidth in a VPC.

# 2 Notes

---

Before using ECS through APIs, learn how to call ECS APIs. For details about how to call REST APIs, see section [API Usage Guidelines](#).

# 3 Request Format

Public cloud APIs follow RESTful API design rules.

Representational State Transfer (REST) allocates Uniform Resource Identifiers (URIs) to dispersed resources so that resources can be located. Applications on clients use Uniform Resource Locators (URLs) to obtain resources.

URLs are in the format of `https://Endpoint/uri`. **Table 3-1** describes the parameters in a URL.

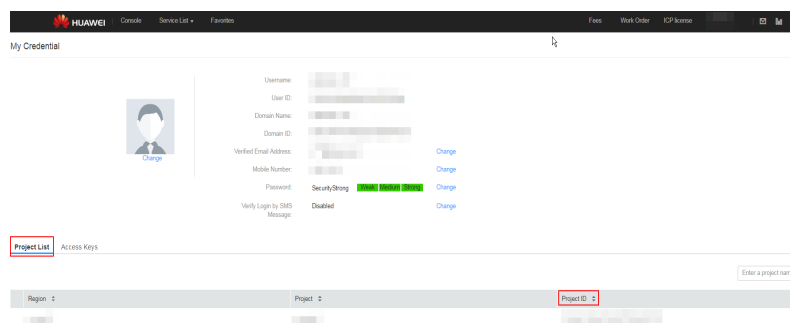
**Table 3-1** Parameter description

Parameter	Description
Endpoint	Specifies the URL that is the entry point for a web service. Obtain the endpoint from <b>Regions and Endpoints</b> .
URI	Specifies a resource path, which is an API access path and obtained from the API URI module, for example, <code>/v2/{tenant_id}/servers</code> .

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID on the console before calling an API. To do so, perform the following operations:

1. Register yourself on the management console and log in to it.
2. Hover the mouse over the username and select **Basic Information** from the drop-down list.
3. On the **Account Info** page, click **Manage** following **Security Credentials**.
4. On the **My Credential** page, view the project ID in the project list.

**Figure 3-1** Viewing project IDs



## 3.1 Versions

OpenStack APIs are of version v2 or v2.1. The two versions use the same set of main code, and v2.1 supports microversions.

## 3.2 Microversions

v2.1 APIs support microversions for small/medium API changes or document changes.

Users can use a microversion to obtain the latest API microversion supported by a cloud service. A cloud service that has been upgraded to the latest microversion is compatible with the original microversions. Users can also use a microversion to obtain new cloud service properties.

The version API returns the minimum and maximum microversions. The client uses the two values to specify the microversion range supported by an API.

### Microversion Response Example

If the values of **version** and **min\_version** are null, the endpoint does not support microversions.

- **version**: indicates the maximum microversion.
- **min\_version**: indicates the minimum microversion.

A microversion on the client must be within the range specified by **version** and **min\_version** to access the endpoint. The client uses the following HTTP header to specify a microversion:

```
X-OpenStack-Nova-API-Version: 2.4
```

Since microversion 2.27, the client can also use the following header to specify a microversion:

```
Openstack-API-Version: compute 2.27
```

In the following response example, the maximum microversion is 2.14 and the minimum one is 2.1:

```
{
  "versions": [
    {
      "id": "v2.0",
      "links": [
        {
          "href": "http://openstack.example.com/v2/",
          "rel": "self"
        }
      ],
      "status": "SUPPORTED",
      "version": "",
      "min_version": "",
      "updated": "2011-01-21T11:33:21Z"
    },
    {
      "id": "v2.1",
      "links": [
        {
          "href": "http://openstack.example.com/v2.1/",

```

```

        "rel": "self"
      }
    ],
    "status": "CURRENT",
    "version": "2.14",
    "min_version": "2.1",
    "updated": "2013-07-23T11:33:21Z"
  }
]
}

```

## Microversion Request Example

For example, you are required to use the API for details about an ECS to view the **OS-EXT-SRV-ATTR:hostname** field.

- **Using a v2 API without a microversion**

- GET: `https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/servers/detail`

Obtain the endpoint from [Regions and Endpoints](#).

- Headers

Content-Type	application/json
X-Auth-Token	\$(token)

- Response body

```

{
  "servers": [
    {
      "tenant_id": "74610f3a5ad941998e91f076297ecf27",
      "addresses": {
        "05d4fb93-84e5-4964-853b-32992ffef627": [
          {
            "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:20:17:95",
            "OS-EXT-IPS:type": "fixed",
            "addr": "192.168.0.228",
            "version": 4
          },
          {
            "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:20:17:95",
            "OS-EXT-IPS:type": "floating",
            "addr": "192.168.51.61",
            "version": 4
          }
        ]
      },
      "metadata": {},
      "OS-EXT-STS:task_state": null,
      "OS-DCF:diskConfig": "MANUAL",
      "OS-EXT-AZ:availability_zone": "eu-de-01",
      "links": [
        {
          "rel": "self",
          "href": "https://None/v2.1/74610f3a5ad941998e91f076297ecf27/servers/89c312bb-285a-4026-a237-d441908c2f9e"
        },
        {
          "rel": "bookmark",
          "href": "https://None/74610f3a5ad941998e91f076297ecf27/servers/89c312bb-285a-4026-a237-d441908c2f9e"
        }
      ],
      "OS-EXT-STS:power_state": 1,
      "id": "89c312bb-285a-4026-a237-d441908c2f9e",

```



```

"os-extended-volumes:volumes_attached": [
  {
    "id": "c70c4b8e-33bd-4d1f-ab16-14a5a38cdeaf"
  }
],
"OS-EXT-SRV-ATTR:host": "pod05.eude01",
"image": {
  "links": [
    {
      "rel": "bookmark",
      "href": "https://None/74610f3a5ad941998e91f076297ecf27/
images/1189efbf-d48b-46ad-a823-94b942e2a000"
    }
  ],
  "id": "1189efbf-d48b-46ad-a823-94b942e2a000"
},
"OS-SRV-USG:terminated_at": null,
"accessIPv4": "",
"accessIPv6": "",
"created": "2018-05-11T03:21:56Z",
"hostId":
"fc7a8ff86bac050f0d9454b1b078dcc97060e819acbf06f04c3e338f",
"OS-EXT-SRV-ATTR:hypervisor_hostname": "nova012e7",
"key_name": "id_rsa",
"flavor": {
  "links": [
    {
      "rel": "bookmark",
      "href": "https://None/74610f3a5ad941998e91f076297ecf27/
flavors/s3.small.1"
    }
  ],
  "id": "s3.small.1"
},
"security_groups": [
  {
    "name": "default"
  }
],
"config_drive": "",
"OS-EXT-STS:vm_state": "active",
"OS-EXT-SRV-ATTR:instance_name": "instance-0016c624",
"user_id": "f79791beca3c48159ac2553fff22e166",
"name": "zt-test",
"progress": 0,
"OS-SRV-USG:launched_at": "2018-05-11T03:22:16.701600",
"updated": "2018-05-11T03:22:51Z",
"status": "ACTIVE"
}
]
}

```

- Conclusion: The response body does not contain the **OS-EXT-SRV-ATTR:hostname** field.

- **Using a v2.1 API with a microversion**

- GET: `https://{endpoint}/v2.1/74610f3a5ad941998e91f076297ecf27/servers/detail`  
Obtain the endpoint from [Regions and Endpoints](#).
- Headers

Content-Type	application/json
X-Auth-Token	\$(token)
X-OpenStack-Nova-API-Version	2.26

## - Response body

```
{
  "servers": [
    {
      "tenant_id": "74610f3a5ad941998e91f076297ecf27",
      "addresses": {
        "05d4fb93-84e5-4964-853b-32992ffef627": [
          {
            "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:20:17:95",
            "OS-EXT-IPS:type": "fixed",
            "addr": "192.168.0.228",
            "version": 4
          },
          {
            "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:20:17:95",
            "OS-EXT-IPS:type": "floating",
            "addr": "192.168.51.61",
            "version": 4
          }
        ]
      },
      "metadata": {},
      "OS-EXT-STS:task_state": null,
      "description": "zt-test",
      "OS-EXT-SRV-ATTR:hostnames": "zt-test",
      "OS-DCF:diskConfig": "MANUAL",
      "OS-EXT-AZ:availability_zone": "eu-de-01",
      "links": [
        {
          "rel": "self",
          "href": "https://None/v2.1/74610f3a5ad941998e91f076297ecf27/servers/89c312bb-285a-4026-a237-d441908c2f9e"
        },
        {
          "rel": "bookmark",
          "href": "https://None/74610f3a5ad941998e91f076297ecf27/servers/89c312bb-285a-4026-a237-d441908c2f9e"
        }
      ],
      "OS-EXT-STS:power_state": 1,
      "id": "89c312bb-285a-4026-a237-d441908c2f9e",
      "os-extended-volumes:volumes_attached": [
        {
          "delete_on_termination": true,
          "id": "c70c4b8e-33bd-4d1f-ab16-14a5a38cdeaf"
        }
      ],
      "locked": false,
      "OS-EXT-SRV-ATTR:kernel_id": "",
      "OS-EXT-SRV-ATTR:host": "pod05.eude01",
      "OS-EXT-SRV-ATTR:ramdisk_id": "",
      "image": {
        "links": [
          {
            "rel": "bookmark",
            "href": "https://None/74610f3a5ad941998e91f076297ecf27/images/1189efbf-d48b-46ad-a823-94b942e2a000"
          }
        ]
      },
      "id": "1189efbf-d48b-46ad-a823-94b942e2a000"
    },
    {
      "accessIPv4": "",
      "OS-SRV-USG:terminated_at": null,
      "accessIPv6": "",
      "OS-EXT-SRV-ATTR:launch_index": 0,
      "created": "2018-05-11T03:21:56Z",
      "OS-EXT-SRV-ATTR:user_data": null,
      "hostId":
      "fc7a8ff86bac050f0d9454b1b078dcc97060e819acbf06f04c3e338f",
    }
  ]
}
```

```
"OS-EXT-SRV-ATTR:reservation_id": "r-pbqmaxer",
"OS-EXT-SRV-ATTR:root_device_name": "/dev/vda",
"host_status": "UP",
"OS-EXT-SRV-ATTR:hypervisor_hostname": "nova01207",
"tags": [],
"key_name": "id_rsa",
"flavor": {
  "links": [
    {
      "rel": "bookmark",
      "href": "https://None/74610f3a5ad941998e91f076297ecf27/
flavors/s3.small.1"
    }
  ],
  "id": "s3.small.1"
},
"security_groups": [
  {
    "name": "default"
  }
],
"config_drive": "",
"OS-EXT-STS:vm_state": "active",
"OS-EXT-SRV-ATTR:instance_name": "instance-0016c624",
"user_id": "f79791beca3c48159ac2553fff22e166",
"name": "zt-test",
"progress": 0,
"OS-SRV-USG:launched_at": "2018-05-11T03:22:16.701600",
"updated": "2018-05-11T03:22:51Z",
"status": "ACTIVE"
}
]
```

- Conclusion: The response body contains the **OS-EXT-SRV-ATTR:hostname** field.

## 3.3 Request Example

**Step 1** Obtain a token.

- POST: <https://{endpoint}/v3/auth/tokens>
- Headers

Content-Type	application/json
--------------	------------------

- Body

```
{
  "auth": {
    "identity": {
      "password": {
        "user": {
          "name": "testuser",
          "domain": {
            "id": "2aa29cbca17a4822abd096610e378ffa"
          }
        },
        "password": "Test@123"
      }
    },
    "methods": [
      "password"
    ]
  },
  "scope": {
    "project": {
      "id": "fb770eb43f934b5a8bda955642b954b9"
    }
  }
}
```

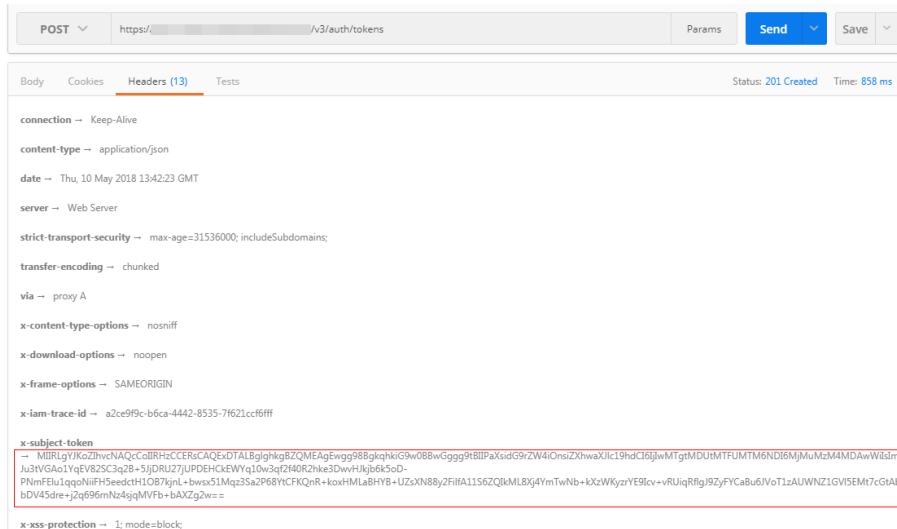
```

}
}
}
}
}

```

Obtain the token (**x-subject-token**) in Headers

Figure 3-2 Obtaining the token



**Step 2** Initiate a request and access the API for viewing details about an ECS.

- GET: `https://{endpoint}/v2/fb770eb43f934b5a8bda955642b954b9/servers/detail`
  - URI format: `GET /v2/{tenant_id}/servers/detail{?changes-since,image,flavor,name,status,limit,marker,not-tags,reservation_id,all_tenants}`
  - `tenant_id`: fb770eb43f934b5a8bda955642b954b9
  - The fields, **changes-since,image,flavor,name,status,limit,marker,not-tags,reservation\_id,all\_tenants**, following the question mark (?) are optional for viewing an ECS.
- Headers

Content-Type	application/json
X-Auth-Token	Obtained in <a href="#">Step 1</a>

- Response body: JSON data in UTF-8 code format

```

{
  "servers": [
    {
      "tenant_id": "fb770eb43f934b5a8bda955642b954b9",
      "addresses": {
        "196b63ba-4201-4b55-b5aa-62ab6085d884": [
          {
            "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:82:2b:a0",
            "OS-EXT-IPS:type": "fixed",
            "addr": "192.168.1.10",
            "version": 4
          },
          {
            "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:82:2b:a0",
            "OS-EXT-IPS:type": "floating",
            "addr": "192.168.213.134",

```

```
        "version": 4
      }
    ]
  },
  "metadata": {},
  "OS-EXT-STS:task_state": null,
  "OS-DCF:diskConfig": "MANUAL",
  "OS-EXT-AZ:availability_zone": "eu-de-02",
  "links": [
    {
      "rel": "self",
      "href": "https://xxx/v2/fb770eb43f934b5a8bda955642b954b9/servers/0e56e372-31d4-40b6-8c85-d82a3cf05c"
    },
    {
      "rel": "bookmark",
      "href": "https://xxx/fb770eb43f934b5a8bda955642b954b9/servers/0e56e372-31d4-40b6-8c85-d82a3cf05c"
    }
  ],
  "OS-EXT-STS:power_state": 1,
  "id": "0e56e372-31d4-40b6-8c85-d82a3cf05c",
  "os-extended-volumes:volumes_attached": [
    {
      "id": "3e3fd674-a816-4602-8175-d9b2e20a65d5"
    }
  ],
  "OS-EXT-SRV-ATTR:host": "pod01.eu-de-02",
  "image": {
    "links": [
      {
        "rel": "bookmark",
        "href": "https://xxx/fb770eb43f934b5a8bda955642b954b9/images/f3966520-45ce-45d3-b099-0123d1cd0043"
      }
    ]
  },
  "id": "f3966520-45ce-45d3-b099-0123d1cd0043"
},
"OS-SRV-USG:terminated_at": null,
"accessIPv4": "",
"accessIPv6": "",
"created": "2018-05-10T09:13:29Z",
"hostId": "1ee40e90e4774fc712d7e881d62ac5be9b05c9006504a69b9ab15aa0",
"OS-EXT-SRV-ATTR:hypervisor_hostname": "nova005@7",
"key_name": null,
"flavor": {
  "links": [
    {
      "rel": "bookmark",
      "href": "https://xxx/fb770eb43f934b5a8bda955642b954b9/flavors/s2.small.1"
    }
  ]
},
"id": "s2.small.1"
},
"security_groups": [
  {
    "name": "default"
  }
],
"config_drive": "",
"OS-EXT-STS:vm_state": "active",
"OS-EXT-SRV-ATTR:instance_name": "instance-0009d9c4",
"user_id": "f79791beca3c48159ac2553fff22e166",
"name": "ecs-65a7",
"progress": 0,
"OS-SRV-USG:launched_at": "2018-05-10T12:11:10.803603",
"updated": "2018-05-10T12:11:10Z",
"status": "ACTIVE"
```

```
},
{
  "tenant_id": "fb770eb43f934b5a8bda955642b954b9",
  "addresses": {
    "21bcff3b-3a71-4304-ab62-dad0b305890e": [
      {
        "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:e8:ab:b2",
        "OS-EXT-IPS:type": "fixed",
        "addr": "192.168.0.79",
        "version": 4
      },
      {
        "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:e8:ab:b2",
        "OS-EXT-IPS:type": "floating",
        "addr": "192.168.218.86",
        "version": 4
      }
    ]
  },
  "metadata": {},
  "OS-EXT-STS:task_state": null,
  "OS-DCF:diskConfig": "MANUAL",
  "OS-EXT-AZ:availability_zone": "eu-de-02",
  "links": [
    {
      "rel": "self",
      "href": "https://xxx/v2/fb770eb43f934b5a8bda955642b954b9/servers/3e6388ea-3467-436e-b11f-4dddbc3dd810"
    },
    {
      "rel": "bookmark",
      "href": "https://xxx/fb770eb43f934b5a8bda955642b954b9/servers/3e6388ea-3467-436e-b11f-4dddbc3dd810"
    }
  ],
  "OS-EXT-STS:power_state": 1,
  "id": "3e6388ea-3467-436e-b11f-4dddbc3dd810",
  "os-extended-volumes:volumes_attached": [
    {
      "id": "1bb5c0f6-300d-45c9-81f0-ad41736716de"
    }
  ],
  "OS-EXT-SRV-ATTR:host": "pod01.eu-de-02",
  "image": {
    "links": [
      {
        "rel": "bookmark",
        "href": "https://xxx/fb770eb43f934b5a8bda955642b954b9/images/f1d75ee7-83bc-4e43-81fb-b69b4625fdea"
      }
    ]
  },
  "id": "f1d75ee7-83bc-4e43-81fb-b69b4625fdea"
},
{
  "OS-SRV-USG:terminated_at": null,
  "accessIPv4": "",
  "accessIPv6": "",
  "created": "2018-01-27T10:01:35Z",
  "hostId": "1ee40e90e4774fc712d7e881d62ac5be9b05c9006504a69b9ab15aa0",
  "OS-EXT-SRV-ATTR:hypervisor_hostname": "nova005@7",
  "key_name": null,
  "flavor": {
    "links": [
      {
        "rel": "bookmark",
        "href": "https://xxx/fb770eb43f934b5a8bda955642b954b9/flavors/s2.small.1"
      }
    ]
  },
  "id": "s2.small.1"
}
```

```
    },
    "security_groups": [
      {
        "name": "default"
      }
    ],
    "config_drive": "",
    "OS-EXT-STS:vm_state": "active",
    "OS-EXT-SRV-ATTR:instance_name": "instance-00070c07",
    "user_id": "f79791beca3c48159ac2553fff22e166",
    "name": "ecs-terraformCLI",
    "progress": 0,
    "OS-SRV-USG:launched_at": "2018-05-10T10:19:04.709851",
    "updated": "2018-05-10T10:19:04Z",
    "status": "ACTIVE"
  }
]
}
```

----End

# 4 Creating an ECS

---

## Scenarios

An ECS with EVS disks is required.

An ECS can be created using a disk or image. This section uses an image as an example to describe how to create an ECS.

## Involved APIs

Creating an ECS involves viewing flavors and AZs as well as creating EVS disks. The following APIs are required:

- API for viewing details about ECS flavors
- API for viewing details about images
- API for viewing networks
- API for creating and importing an SSH key pair
- API for creating an ECS
- API for viewing details about an ECS

## Procedure

### Step 1 Determine the ECS flavor.

#### 1. View ECS flavors.

- API

URI format: GET /v2/{tenant\_id}/flavors/detail{?  
minDisk,minRam,is\_public,sort\_key,sort\_dir}

The fields following the question mark (?) are optional for viewing flavors. For details, see section [Querying Details About ECS Flavors](#).

- Request example

GET: https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/flavors/detail

Obtain the endpoint from [Regions and Endpoints](#).

- Response example

```
{  
  "flavors": [  
    {
```



```
    "name": "c1.2xlarge",
    "links": [
      {
        "href": "https://xxx/v2/74610f3a5ad941998e91f076297ecf27/
flavors/c1.2xlarge",
        "rel": "self"
      },
      {
        "href": "https://xxx/74610f3a5ad941998e91f076297ecf27/flavors/
c1.2xlarge",
        "rel": "bookmark"
      }
    ],
    "ram": 8192,
    "OS-FLV-DISABLED:disabled": false,
    "vcpus": 8,
    "swap": "",
    "os-flavor-access:is_public": true,
    "rxtx_factor": 1,
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "disk": 0,
    "id": "c1.2xlarge"
  }
]
}
```

2. Select a flavor based on site requirements and record the flavor ID.

## Step 2 Determine the image.

1. View images.

- API

URI format: GET /v2/{tenant\_id}/images/detail

For details, see section "Querying Image Details" in *Elastic Cloud Server API Reference*.

- Request example

GET: https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/images/detail

Obtain the endpoint from [Regions and Endpoints](#).

- Response example

```
{
  "images": [
    {
      "OS-EXT-IMG-SIZE:size": 0,
      "metadata": {
        "__os_type": "Linux",
        "hw_vif_multiqueue_enabled": "true",
        "__imagetype": "gold",
        "__quick_start": "true",
        "virtual_env_type": "FusionCompute",
        "__support_xen": "true",
        "__support_kvm": "true",
        "__image_source_type": "uds",
        "__platform": "EulerOS",
        "__os_version": "EulerOS 2.2 64bit",
        "__os_bit": "64",
        "__isregistered": "false"
      },
      "created": "2018-05-14T06:13:50Z",
      "minRam": 0,
      "name": "DBS-MySQL-Image_2.1.3.3",
      "progress": 100,
      "links": [
        {
          "rel": "self",
          "href": "https://None/v2/74610f3a5ad941998e91f076297ecf27/"
        }
      ]
    }
  ]
}
```

```
images/11e8f727-d439-4ed1-b3b8-33f46c0379c4"
    },
    {
      "rel": "bookmark",
      "href": "https://None/74610f3a5ad941998e91f076297ecf27/images/
11e8f727-d439-4ed1-b3b8-33f46c0379c4"
    },
    {
      "rel": "alternate",
      "href": "https://None/images/11e8f727-d439-4ed1-
b3b8-33f46c0379c4",
      "type": "application/vnd.openstack.image"
    }
  ],
  "id": "11e8f727-d439-4ed1-b3b8-33f46c0379c4",
  "updated": "2018-05-14T06:13:52Z",
  "minDisk": 40,
  "status": "ACTIVE"
}
]
}
```

2. Select an image based on site requirements and record the image ID.

### Step 3 Determine the network configuration.

1. View networks.

- API

URI format: GET /v2/{tenant\_id}/os-networks

For details, see section "Querying Networks" in *Elastic Cloud Server API Reference*.

- Request example

GET: https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/os-networks

Obtain the endpoint from [Regions and Endpoints](#).

- Response example

```
{
  "networks": [
    {
      "id": "07a9557d-4256-48ae-847c-415a9c8f7ff6",
      "label": "b_tt3_tdlb",
      "broadcast": null,
      "cidr": null,
      "dns1": null,
      "dns2": null,
      "gateway": null,
      "netmask": null,
      "cidr_v6": null,
      "gateway_v6": null,
      "netmask_v6": null
    }
  ]
}
```

2. Select a network based on site requirements and record the network ID.

### Step 4 Set the login mode to key pair.

1. Create a key pair.

- API

URI format: POST /v2/{tenant\_id}/os-keypairs

For details, see section "Creating and Importing an SSH Key Pair" in *Elastic Cloud Server API Reference*.

## - Request example

POST: `https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/os-keypairs`Obtain the endpoint from [Regions and Endpoints](#).

## Body

```
{
  "keypair": {
    "type": "ssh",
    "name": "demo1",
    "user_id": "fake"
  }
}
```

## - Response example

```
{
  "keypair": {
    "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCrR5Gcwlh5ih7JOvzIUuQxS5qzWWPMYHeDXkDKSQ9W5
pumOV05SiO3WCswnaQ5xMdO131mNiHtwlwq9dJi7X6jJBB2shTD
+00G5WuwkBbFU4CLvt1B44u0NUiaTJ35NANvW2/4XvpXm90wiQ3B5ge6ZY7Esi38Unh
+pkbhPkYxNBCK8yoOlojQhWs75abdxZBi811/8RwLcNiFiocA2RGxtRjBdpEScj+1TU
+OcfZdQnr0AFbO11z7yxfIyggwzVTgUuJNbMbKHStQqRbkl1FLHY4RBPQgb7RN/
YaXKTQSXT84k+D9x1LDNo7Wj4fwOJTOz/s/PvbIOqjRHt9D6Y4IKd Generated-by-Nova
\n",
    "private_key": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEogIBAAKCAQEAg0eRnMJYeYoeyTr8yFLkMUas1ljzGB3g15AykkPVuabpjd
\nOUojt1grMJ2kOCTHTpd9ZjYh7cJcKvXSYu1+oyQQdrIUw/tNBuVrsJAWxVOAi77d
\nQeOLtDVIImkyd+TQL1tv+F76V5vTsIkNweYHumWOxLiT/FJ4fqZG4T5GMTQQivMqD
\npaI0IVrO+Wm3cWQYvNdf/EcC3DYhYqHANKRsbUYwXaREnI/tU1Pjnh2XUJ69ABWz\ntdc
+8sXyMoMM1U4FLiTWzGyh0rUKkW5JXzJR2OEQT0IG+0Tf2Glyk0E10/OJPg/\ncZQzaO1o
+H8DiUzs/7Pz72yDqo0R7fQ+mOCCnQIDAQABAOIBAA6/c9dGmK2mae4z
\nyQ5KrOFdvC1TNhej+sZx+CwyzEJUSvSuHcvQCXFBaz8FY92hhvPKcX66jINXZ
+4/\nCmWAQ5YyhcRiow0Y91HvsS0bywknX3q6kxBfodmyyCWFkgd5iMTADb1Lx0a27Y7\nj1
S4Dl5gy1GmxUN2Ng24wWEAje8ZnuI0lrt5IZKp+s5IAi/rb5AG/mL7EziE8c\nmGP+QAa
+nzwhAwNhFwVID230xen/ZcoL1d77hxeARNqJUxoR25gJd6Ebg2y9pDW
\nVu6cbbzgdGUCfQY1MEoAamAkCsw0sDpVDBXwQnt2A537n6Wq2bgYIKusHr9tthxP\n/
5ubQLUCgYEA4zYuBG2vtLHnvc26P8o2j1xcJS9K0ozkah9JF13hqFN0sAqL1z7\n/
FmljA4kzHJS3d0UqP3AMDxY3HkIqCn4Be7lqeAAe2AfkqOZpt9MDnv4VwKe9sPb
\nViW1qjL3FfziLC/YWTRNS1pwRjqJGhA+UQt8rOialk/zXmrEs7bXLCgYEAwPsu
\nK3j5QoAiziYVMYf5iCzWwAM9Ljpf9gw23lefTdIzhhfFtJp1VRSyxRGU0UZ84GMI
\nTd5zmcIF/1KUfhqmeiQzz6NIPEYEReahjPQ/sOH/Gk5Rwr3QwYPrwAu5x+kk/SRi
\nKPkqW7APTR0sMQBcUq+ZyWGYLGPmdd1zUdLfb0sCgYBkuz11iydtxb3G/obSD2WO
\nM9VaIycmzRPFzNwGRH/gOR0mhtluKp0wyJjbSd34oeqPH/2r2ivddrOysxoqa8jg
\n4IQDZyLv7MaKjQxrieqP89+y9Or9TmFo1xB46x2G8EN8/xHuA9YGnZSPFtWv72m
\nhRqV0hv82amWsA0vHnRUSwKBgDsKHXvrTmBnkNkkykMXCH5iyWiBFSyZa1ZJMLgf
\nknsqfdzeVPwF6E55QKAN2uuTlwG/3ljPxahR1hvmUJjQN9JSBiUKbtW6GPCRVbr\nf/
jLi1Iu99COZd1uVKeybqn8Z/asNP24DR9FM8kxzZ1IMPaTBmhFypp6BclhcLbt
\nxTG1AoGAfcrkVbV1SOy7fECUTmpUECcw0yU4Gwj3sR2RbII63C500RVYQ1UpUaRR
\naANbASHTVR4myOKtGSxEUhAQHlxFDwsDL7W3gzAqTFbEDplxAaUyT/
nkOAhQjEm4\nORFdDETeXLQG1KMUj+8AdnhfYp3Jtdft6rmPpZEBUfiCAUMAvb0=\n-----
END RSA PRIVATE KEY-----\n",
    "user_id": "f79791beca3c48159ac2553fff22e166",
    "name": "demo1",
    "fingerprint": "57:a7:a2:ed:5f:aa:e7:54:62:2e:bb:e7:92:22:cb:40"
  }
}
```

## 2. Import the key pair.

## - API

URI format: POST `/v2/{tenant_id}/os-keypairs`For details, see section "Creating and Importing an SSH Key Pair" in *Elastic Cloud Server API Reference*.

## - Request example

POST: `https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/os-keypairs`

Obtain the endpoint from [Regions and Endpoints](#).

Body

```
{
  "keypair": {
    "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDy8wMTdBViJgi62o6eSho0lSKx3CZ3cE6PHisDblfK3
Y0Bg7EHV7iV9c74pqsrlhK0xuGUu01NxDQWbkwLTPN4F9Iy5CIYohLuMIpbln6LDtfrPpdhEh
3lxL8MM6lgyfpKzeKkwkEpSFj27Rgh6zCyJgBpkA2A0HTP737UlitahL4faCWDIS
+Vj6mbcfkWiMhuMCzTZgSKAZ4PfoG4B5HJhR52C6A4XLiQFT9heh9gnIsIG
+uTogTKUbcJKuN7M6AraJpul6eHhV9YI4433sDmuiBF/njvrvVPWwAHLAkgt9I8q1T/
cfEFiwzXpdGbkK508NC7K+qNbbdKihlahONT Generated-by-Nova\n",
    "type": "ssh",
    "name": "demo2",
    "user_id": "fake"
  }
}
```

- Response example

```
{
  "keypair": {
    "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDy8wMTdBViJgi62o6eSho0lSKx3CZ3cE6PHisDblfK3
Y0Bg7EHV7iV9c74pqsrlhK0xuGUu01NxDQWbkwLTPN4F9Iy5CIYohLuMIpbln6LDtfrPpdhEh
3lxL8MM6lgyfpKzeKkwkEpSFj27Rgh6zCyJgBpkA2A0HTP737UlitahL4faCWDIS
+Vj6mbcfkWiMhuMCzTZgSKAZ4PfoG4B5HJhR52C6A4XLiQFT9heh9gnIsIG
+uTogTKUbcJKuN7M6AraJpul6eHhV9YI4433sDmuiBF/njvrvVPWwAHLAkgt9I8q1T/
cfEFiwzXpdGbkK508NC7K+qNbbdKihlahONT Generated-by-Nova\n",
    "user_id": "f79791beca3c48159ac2553fff22e166",
    "name": "demo2",
    "fingerprint": "dd:44:45:49:d9:f6:4f:c0:24:2d:81:aa:c4:4b:83:c2"
  }
}
```

- Record the name in the response body, for example, **demo2**.

### Step 5 Create an ECS authenticated using the key pair.

- API

URI format: POST /v2/{tenant\_id}/servers

For details about API constraints and request parameters, see section "Creating an ECS" in *Elastic Cloud Server API Reference*.

 **NOTE**

In this example, the ECS is created using a specified image. Therefore,

- In **block\_device\_mapping\_v2**, set **source\_type** to **image**, **uuid** to the image ID, **destination\_type** to **volume**, and **boot\_index** to **0**.
- The **volume\_size** must be greater than or equal to the minimum value specified in the image metadata.

- Request example

POST: <https://{{endpoint}}/v2/74610f3a5ad941998e91f076297ecf27/servers>

Obtain the endpoint from [Regions and Endpoints](#).

Body

```
{
  "server": {
    "flavorRef": "c1.large",
    "name": "ztttestvml",
    "block_device_mapping_v2": [{
      "source_type": "image",
      "destination_type": "volume",
      "volume_type": "SATA",
      "volume_size": "40",
      "delete_on_termination": "true",
    }
  ]
}
```

```
"uuid": "11e8f727-d439-4ed1-b3b8-33f46c0379c4",
"boot_index": "0"
}],
"networks": [{
  "uuid": "fb68519f-a7c0-476e-98d4-2e4cf6de6def"
}],
"key_name": "demo2",
"availability_zone": "eu-de-01"
}
}
```

- Response example

```
{
  "server": {
    "security_groups": [
      {
        "name": "default"
      }
    ],
    "OS-DCF:diskConfig": "MANUAL",
    "links": [
      {
        "rel": "self",
        "href": "https://None/v2/74610f3a5ad941998e91f076297ecf27/servers/6d311127-bce1-48db-bf0f-cac9f8f7f077"
      },
      {
        "rel": "bookmark",
        "href": "https://None/74610f3a5ad941998e91f076297ecf27/servers/6d311127-bce1-48db-bf0f-cac9f8f7f077"
      }
    ],
    "id": "6d311127-bce1-48db-bf0f-cac9f8f7f077",
    "adminPass": "WcC4QoVZPXpV"
  }
}
```

### Step 6 Verify the ECS creation.

- API

URI format: GET /v2/{tenant\_id}/servers/{server\_id}

For details, see section "Querying Details About an ECS" in *Elastic Cloud Server API Reference*.

- Request example

GET: <https://endpoint/v2/74610f3a5ad941998e91f076297ecf27/servers/0c71c0da-8852-4c56-a1d1-3a9b9bcb6da6>

Where,

**0c71c0da-8852-4c56-a1d1-3a9b9bcb6da6** is the UUID of the created ECS.

Obtain the endpoint from [Regions and Endpoints](#).

- Response example

```
{
  "server": {
    "tenant_id": "74610f3a5ad941998e91f076297ecf27",
    "addresses": {
      "2a6f4aa6-d93e-45f5-a8cb-b030dbf8cd68": [
        {
          "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:88:01:1b",
          "OS-EXT-IPS:type": "fixed",
          "addr": "192.168.2.192",
          "version": 4
        }
      ]
    },
    "metadata": {},
  }
}
```

```
"OS-EXT-STS:task_state": null,
"OS-DCF:diskConfig": "MANUAL",
"OS-EXT-AZ:availability_zone": "eu-de-01",
"links": [
  {
    "rel": "self",
    "href": "https://None/v2/74610f3a5ad941998e91f076297ecf27/servers/0c71c0da-8852-4c56-ald1-3a9b9bcb6da6"
  },
  {
    "rel": "bookmark",
    "href": "https://None/74610f3a5ad941998e91f076297ecf27/servers/0c71c0da-8852-4c56-ald1-3a9b9bcb6da6"
  }
],
"OS-EXT-STS:power_state": 1,
"id": "0c71c0da-8852-4c56-ald1-3a9b9bcb6da6",
"os-extended-volumes:volumes_attached": [
  {
    "id": "b551445a-e749-4d53-932a-638a455cb6c3"
  }
],
"OS-EXT-SRV-ATTR:host": "pod1a.eud1",
"image": {
  "links": [
    {
      "rel": "bookmark",
      "href": "https://None/74610f3a5ad941998e91f076297ecf27/images/11e8f727-d439-4ed1-b3b8-33f46c0379c4"
    }
  ],
  "id": "11e8f727-d439-4ed1-b3b8-33f46c0379c4"
},
"OS-SRV-USG:terminated_at": null,
"accessIPv4": "",
"accessIPv6": "",
"created": "2018-05-25T01:47:11Z",
"hostId": "b2792bef989888d2df1f51bff81de5ac58a4117f4e9ec3059c1a0410",
"OS-EXT-SRV-ATTR:hypervisor_hostname": "nova001@36",
"key_name": null,
"flavor": {
  "links": [
    {
      "rel": "bookmark",
      "href": "https://None/74610f3a5ad941998e91f076297ecf27/flavors/c1.large"
    }
  ],
  "id": "c1.large"
},
"security_groups": [
  {
    "name": "default"
  }
],
"config_drive": "",
"OS-EXT-STS:vm_state": "active",
"OS-EXT-SRV-ATTR:instance_name": "instance-001883cd",
"user_id": "f79791beca3c48159ac2553fff22e166",
"name": "zttestvml",
"progress": 0,
"OS-SRV-USG:launched_at": "2018-05-25T01:47:55.755922",
"updated": "2018-05-25T01:47:55Z",
"status": "ACTIVE"
}
}
```

----End

# 5 Modifying ECS Specifications

---

## Scenarios

When ECS specifications fail to meet service requirements, they can be modified, for example, by upgrading the vCPUs and memory. Certain ECSs also support changing ECS types.

## Constraints

- You can modify the ECS specifications only when the ECS is stopped.
- The EVS disk capacity of the ECS cannot be reduced during the specifications modification.
- ECS specifications (vCPU or memory) reduction degrades the ECS performance.
- Certain ECSs do not support specifications modification. For details about available ECS types as well as their functions and usage on the public cloud, see section "Notes" in [Instances and Application Scenarios](#).

## Involved APIs

Modifying ECS specifications involves the following APIs:

- API for modifying the specifications of an ECS
- API for confirming ECS specifications modification
- API for rolling back ECS specifications modification

## Procedure

**Step 1** Modify the ECS specifications.

- API

URI format: POST /v2/{tenant\_id}/servers/{server\_id}/action

For details, see section "Modifying the Specifications of an ECS" in *Elastic Cloud Server API Reference*.

- Request example

```
{
  "resize": {
    "flavorRef": "4"
  }
}
```

- Response example  
N/A

**Step 2** Confirm the specifications modification.

The ECS must be in **resized** state. That is, the **OS-EXT-STS:vm\_state** value of the ECS must be **resized**.

- API

URI format: POST /v2/{tenant\_id}/servers/{server\_id}/action

For details, see section "Confirming ECS Specifications Modification" in *Elastic Cloud Server API Reference*.

- Request example

```
{  
  "confirmResize": null  
}
```

- Response example  
N/A

**Step 3** (Optional) Roll back the specifications modification.

Fallback notice:

The ECS must be in **resized** state. That is, the **OS-EXT-STS:vm\_state** value of the ECS must be **resized**.

The data modified during specifications modification will be lost after the rollback.

- API

URI format: POST /v2/{tenant\_id}/servers/{server\_id}/action

For details, see section "Rolling Back ECS Specifications Modification" in *Elastic Cloud Server API Reference*.

- Request example

```
{  
  "revertResize": null  
}
```

- Response example  
N/A

----End



# 6 Attaching a Disk to an ECS

## Scenarios

If the existing disks of an ECS fail to meet service requirements, for example, due to insufficient disk space or poor disk performance, you can attach more available disks to the ECS, or call the EVS disk creation API to create disks and attach them to the ECS. To attach an EVS disk to an ECS, you need to call the required API.

### NOTE

You can attach a data disk by setting the **data\_volumes** parameter during ECS creation or attach a data disk after the ECS is created. This section describes how to attach a disk to a created ECS.

## Involved APIs

Attaching a disk involves the following APIs:

- API for creating an EVS disk
- API for attaching a disk to an ECS
- API for viewing disks attached to an ECS

## Procedure

### Step 1 Create an EVS disk.

#### 1. Create an EVS disk.

- API

URI format: `POST /v2/{tenant_id}/volumes`

For details, see section "Creating an EVS Disk (Native OpenStack API v2)" in *Elastic Cloud Server API Reference*.

- Request example

POST: `https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/volumes`

Obtain the endpoint from [Regions and Endpoints](#).

Body

```
{
  "volume": {
    "name": "openapi_vol02",
    "availability_zone": "eu-de-01",
    "description": "create for api test",
```

```
    "volume_type": "SATA",
    "size": 40
  }
}
```

- **Response example**

```
{
  "volume": {
    "status": "creating",
    "user_id": "f79791beca3c48159ac2553fff22e166",
    "attachments": [],
    "links": [
      {
        "href": "https://xxx/v2/74610f3a5ad941998e91f076297ecf27/volumes/51f45e08-1d4f-44c6-a4a9-84a488e0e8d3",
        "rel": "self"
      },
      {
        "href": "https://xxx/74610f3a5ad941998e91f076297ecf27/volumes/51f45e08-1d4f-44c6-a4a9-84a488e0e8d3",
        "rel": "bookmark"
      }
    ],
    "availability_zone": "eu-de-01",
    "bootable": "false",
    "encrypted": false,
    "created_at": "2018-05-16T11:19:33.992984",
    "description": "create for api test",
    "updated_at": null,
    "volume_type": "SATA",
    "name": "openapi_vol02",
    "replication_status": "disabled",
    "consistencygroup_id": null,
    "source_volid": null,
    "snapshot_id": null,
    "shareable": false,
    "multiattach": false,
    "metadata": {
      "__system__volume_name": "openapi_vol02"
    },
    "id": "51f45e08-1d4f-44c6-a4a9-84a488e0e8d3",
    "size": 40
  }
}
```

2. Record the **volume ID** in the response.

## Step 2 Attach the disk to the ECS.

- API

URI format: POST /v2/{tenant\_id}/servers/{server\_id}/os-volume\_attachments

For details, see section "Attaching a Disk to an ECS" in *Elastic Cloud Server API Reference*.

- Request example

`https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/servers/9f4d9281-95e7-4915-a126-1ee597101e2e/os-volume_attachments`

Obtain the endpoint from [Regions and Endpoints](#).

Body

```
{
  "volumeAttachment": {
    "volumeId": "51f45e08-1d4f-44c6-a4a9-84a488e0e8d3",
    "device": "/dev/sdb"
  }
}
```

- Response example

```
{
  "volumeAttachment": {
    "id": "51f45e08-1d4f-44c6-a4a9-84a488e0e8d3",
    "volumeId": "51f45e08-1d4f-44c6-a4a9-84a488e0e8d3",
    "serverId": "9f4d9281-95e7-4915-a126-1ee597101e2e",
    "device": "/dev/sdb"
  }
}
```

**Step 3** Verify the disk attachment.

- API

URI format: GET /v2/{tenant\_id}/servers/{server\_id}/os-volume\_attachments

For details, see section "Querying Disks Attached to an ECS" in *Elastic Cloud Server API Reference*.

- Request example

`https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/servers/9f4d9281-95e7-4915-a126-1ee597101e2e/os-volume_attachments`

Obtain the endpoint from [Regions and Endpoints](#).

- Response example

```
{
  "volumeAttachments": [
    {
      "volumeId": "4fc0b4cc-9d6c-431c-be70-3dfeec2ff6e0",
      "id": "4fc0b4cc-9d6c-431c-be70-3dfeec2ff6e0",
      "device": "/dev/sda",
      "serverId": "9f4d9281-95e7-4915-a126-1ee597101e2e"
    },
    {
      "volumeId": "51f45e08-1d4f-44c6-a4a9-84a488e0e8d3",
      "id": "51f45e08-1d4f-44c6-a4a9-84a488e0e8d3",
      "device": "/dev/sdb",
      "serverId": "9f4d9281-95e7-4915-a126-1ee597101e2e"
    }
  ]
}
```

**---End**

# 7 Binding a NIC to an ECS

---

## Scenarios

If an ECS requires multiple NICs, you can call the API for creating NICs and bind them to the ECS.

 **NOTE**

You can bind a NIC by setting the **nics** parameter during ECS creation or bind a NIC after the ECS is created. This section describes how to bind a NIC to a created ECS.

## Involved APIs

Binding a NIC involves the following APIs:

- API for creating a network
- API for creating a subnet
- API for creating a port
- API for binding a NIC to an ECS
- API for viewing ECS NICs

## Procedure

### Step 1 Create a NIC.

1. Create a network.
  - API  
URI format: POST /v2.0/networks  
For details, see section "Creating a Network" in *Virtual Private Cloud API Reference*.
  - Request example  
POST: `https://{endpoint}/v2.0/networks`  
Obtain the endpoint from [Regions and Endpoints](#).  
Body

```
{
  "network": {
    "shared": false,
    "name": "demo-net",
```

```
    "admin_state_up": true,  
    "tenant_id": "74610f3a5ad941998e91f076297ecf27"  
  }  
}
```

- Response example

```
{  
  "network": {  
    "id": "c4a3019d-1ac0-4cfb-a838-2342eb992e6b",  
    "name": "demo-net",  
    "status": "ACTIVE",  
    "shared": false,  
    "subnets": [],  
    "availability_zone_hints": [],  
    "availability_zones": [  
      "eu-de-01",  
      "eu-de-02"  
    ],  
    "admin_state_up": true,  
    "tenant_id": "74610f3a5ad941998e91f076297ecf27",  
    "provider:network_type": "vxlan",  
    "router:external": false  
  }  
}
```

2. Record the **network** ID in the response.

3. Create a subnet.

- API

URI format: POST /v2.0/subnets

For details, see section "Creating a Subnet" in *Virtual Private Cloud API Reference*.

- Request example

POST: https://{endpoint}/v2.0/subnets

Obtain the endpoint from [Regions and Endpoints](#).

Body

```
{  
  "subnet": {  
    "name": "testsubnet",  
    "enable_dhcp": true,  
    "network_id": "c4a3019d-1ac0-4cfb-a838-2342eb992e6b",  
    "tenant_id": "74610f3a5ad941998e91f076297ecf27",  
    "dns_nameservers": [  
      "8.8.8.8",  
      "8.8.8.7"  
    ],  
    "allocation_pools": [  
      {  
        "start": "10.0.10.2",  
        "end": "10.0.10.254"  
      }  
    ],  
    "host_routes": [],  
    "ip_version": 4,  
    "gateway_ip": "10.0.10.1",  
    "cidr": "10.0.10.0/24"  
  }  
}
```

- Response example

```
{  
  "subnet": {  
    "name": "testsubnet",  
    "cidr": "10.0.10.0/24",  
    "id": "877b5567-e8c6-4a0d-aabf-0f13da225fe5",  
    "enable_dhcp": true,  
    "network_id": "c4a3019d-1ac0-4cfb-a838-2342eb992e6b",  
  }  
}
```

```
"tenant_id": "74610f3a5ad941998e91f076297ecf27",
"dns_nameservers": [
  "8.8.8.8",
  "8.8.8.7"
],
"allocation_pools": [
  {
    "start": "10.0.10.2",
    "end": "10.0.10.254"
  }
],
"host_routes": [],
"ip_version": 4,
"gateway_ip": "10.0.10.1"
}
```

4. Record the **subnet ID** in the response.

5. Create a port.

- API

URI format: POST /v2.0/ports

For details, see section "Creating a Port" in *Virtual Private Cloud API Reference*.

- Request example

POST: <https://{{endpoint}}/v2.0/ports>

Obtain the endpoint from [Regions and Endpoints](#).

Body

```
{
  "port": {
    "admin_state_up": true,
    "fixed_ips": [
      {
        "subnet_id": "877b5567-e8c6-4a0d-aabf-0f13da225fe5"
      }
    ],
    "name": "test",
    "network_id": "c4a3019d-1ac0-4cfb-a838-2342eb992e6b",
    "tenant_id": "74610f3a5ad941998e91f076297ecf27"
  }
}
```

- Response example

```
{
  "port": {
    "id": "7bf1c36f-e7f8-478a-be3d-674b486abbc4",
    "name": "test",
    "status": "DOWN",
    "admin_state_up": true,
    "fixed_ips": [
      {
        "subnet_id": "877b5567-e8c6-4a0d-aabf-0f13da225fe5",
        "ip_address": "10.0.10.233"
      }
    ],
    "mac_address": "fa:16:3e:db:91:f6",
    "network_id": "c4a3019d-1ac0-4cfb-a838-2342eb992e6b",
    "tenant_id": "74610f3a5ad941998e91f076297ecf27",
    "device_id": "",
    "device_owner": "",
    "security_groups": [
      "93031677-2895-4b83-855a-637e309aa9e6"
    ],
    "extra_dhcp_opts": [],
    "allowed_address_pairs": []
  }
}
```

```
"binding:vnic_type": "normal",
"binding:vif_details": {},
"binding:profile": {}
}
```

- Record the **port ID** in the response.

## Step 2 Bind the NIC to the ECS.

- API

URI format: POST /v2/{tenant\_id}/servers/{server\_id}/os-interface

For details, see section "Adding a NIC to an ECS" in *Elastic Cloud Server API Reference*.

- Request example

POST: <https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/servers/9f4d9281-95e7-4915-a126-1ee597101e2e/os-interface>

Obtain the endpoint from [Regions and Endpoints](#).

Body

```
{
  "interfaceAttachment": {
    "port_id": "7bf1c36f-e7f8-478a-be3d-674b486abbc4"
  }
}
```

- Response example

```
{
  "interfaceAttachment": {
    "port_state": "ACTIVE",
    "fixed_ips": [
      {
        "subnet_id": "877b5567-e8c6-4a0d-aabf-0f13da225fe5",
        "ip_address": "10.0.10.233"
      }
    ],
    "port_id": "7bf1c36f-e7f8-478a-be3d-674b486abbc4",
    "net_id": "c4a3019d-1ac0-4cfb-a838-2342eb992e6b",
    "mac_addr": "fa:16:3e:db:91:f6"
  }
}
```

## Step 3 Verify the NIC binding.

- API

URI format: GET /v2/{tenant\_id}/servers/{server\_id}/os-interface

For details, see section "Querying ECS NICs" in *Elastic Cloud Server API Reference*.

- Request example

GET: <https://{endpoint}/v2/74610f3a5ad941998e91f076297ecf27/servers/9f4d9281-95e7-4915-a126-1ee597101e2e/os-interface>

Obtain the endpoint from [Regions and Endpoints](#).

- Response example

```
{
  "interfaceAttachments": [
    {
      "port_state": "ACTIVE",
      "fixed_ips": [
        {
          "subnet_id": "46712fe4-25bd-4eae-874b-a528abfb76be",
          "ip_address": "192.168.0.50"
        }
      ]
    }
  ],
}
```

```
    "port_id": "dd706739-b696-40be-a9f4-477ce478cb18",  
    "net_id": "17251a8f-a671-4d7c-85d9-af5415962994",  
    "mac_addr": "fa:16:3e:a5:e0:3c"  
  },  
{  
  "port_state": "ACTIVE",  
  "fixed_ips": [  
    {  
      "subnet_id": "877b5567-e8c6-4a0d-aabf-0f13da225fe5",  
      "ip_address": "10.0.10.233"  
    }  
  ],  
  "port_id": "7bf1c36f-e7f8-478a-be3d-674b486abbc4",  
  "net_id": "c4a3019d-1ac0-4cfb-a838-2342eb992e6b",  
  "mac_addr": "fa:16:3e:db:91:f6"  
}  
]  
}
```

----End



# A Change History

---

Release Date	What's New
2018-11-08	This issue is the first official release.