SDK

Development Guide

Issue	10
Date	2020-05-18

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 SDK Overview

A software development kit (SDK) contains code and examples that you use to create cloud applications in the language of your choice.

Currently, the SDK supports Java, Go, and Python languages. If one of the following SDKs do not support your language or examples, you can use the APIs or any other known SDKs.

2 Java

2.1 Getting Started with Java SDK

Welcome to use HUAWEI CLOUD developer tools (Java SDK). Java SDK allows you to easily access cloud services using codes.

This tutorial describes how to install and use Java SDK and provides examples to help you quickly get started.

The supported Java SDK is developed based on OpenStack4j.

Supported Cloud Services

The Java SDK supports the following cloud services:

- **Identity and Access Management (IAM)**
- **Image Management Service (IMS)**
- **Virtual Private Cloud (VPC)**
- **Elastic Cloud Server (ECS)**
- **Elastic Volume Service (EVS)**
- **Auto Scaling (AS)**
- **Cloud Eye (CES)**
- **Domain Name Service (DNS)**
- **Elastic Load Balancing (ELB)**
- **Volume Backup Service (VBS)**
- **Cloud Trace Service (CTS)**
- **Key Management System (KMS)**
- **Anti-DDoS**
- **Distributed Message Service (DMS)**
- **MapReduce Service (MRS)**
- **Content Delivery Network (CDN)**
- **Tag Management Service (TMS)**

- **Enterprise Management (EPS)**
- **Relational Database Service (RDS)**

Prerequisites

1. You have obtained a cloud platform account and provisioned all required services.
2. You have installed JDK. The Java SDK is applicable to JDK1.8 and later versions. You are advised to use JDK1.8.

SDK Acquisition and Installation

Add the following Maven dependency to the **pom.xml** file to install the Java SDK:

```
<dependency>
<groupId>com.huawei</groupId>
<artifactId>openstack4j</artifactId>
<version>1.0.12</version>
</dependency>
```

For the latest version supported by Java SDK, see **here**.

The services involved in this document use the same JAR file.

How to Use

Set parameters, initialize the SDK client, and invoke the SDK to access the service API. For details about the parameters, see **Table 2-1**.

```
package demo;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import com.huawei.openstack4j.openstack.OSFactory;
import com.huawei.openstack4j.api.OSClient.OSClientV3;
import com.huawei.openstack4j.core.transport.Config;
import com.huawei.openstack4j.model.common.Identifier;
import com.huawei.openstack4j.model.compute.Server;

public class Demo {
    public static void main(String[] args) {
        // Set the authentication parameters.
        String authUrl = "https://iam.example.com/v3";//endpointUrl
        String user = "replace-with-your-username";//username
        String password = "replace-with-your-password";//user password
        String projectId = "replace-with-your-projectId";//project ID
        String userDomainId = "replace-with-your-domainId";//account ID

        // Initialize the client.
        OSClientV3 os = OSFactory.builderV3()
                .endpoint(authUrl)
                .credentials(user, password, Identifier.byId(userDomainId))
                .scopeToProject(Identifier.byId(projectId)).authenticate();

        // Set query parameters.
        Map<String, String> filter = new HashMap<String, String>();

        // Put the parameters that need to be entered into the filter.
        filter.put("limit", "3");

        // Invoke the interface for querying the VM List.
        List<? extends Server> serverList = os.compute().servers().list(filter);
```



```
        if (serverList.size() > 0) {
            System.out.println("get serverList success, size = " + serverList.size());
            for (Server server : serverList) {
                System.out.println(server);
            }
        } else {
            System.out.println("no server exists.");
        }
    }
}
```

Table 2-1 Parameter description

Parameter	Description	Example Value
authUrl	Specifies the endpoint of the IAM service. **example** in https://iam.*example*.com/v3 indicates **Region.Cloud platform domain name**. For details about the parameter, see **here**.	https://iam.cn-north-1.myhuaweicloud.com/v3
user	Specifies the IAM username. For details about how to obtain the username, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
password	Specifies the IAM user password.	N/A
projectId	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	None
userDomainId	Specifies the account ID. For details about how to obtain the account ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	None

2.2 Using the Java SDK

2.2.1 Installing the Java SDK

Java SDK can be installed by importing Java SDK JAR file, adding Maven dependency, or downloading Java SDK from HUAWEI CLOUD mirror center.

Method 1: Importing Java SDK JAR File

You can download the JAR file from the GitHub website and import the package to the interactive development environment (IDE).

https://github.com/huaweicloud/huaweicloud-sdk-release/tree/master/java-sdk

Take Eclipse as an example. After creating a Java project, perform the following steps to import the JAR file to the new project:

1. Copy the downloaded JAR file to the project folder.
2. Open the project in Eclipse, right-click the project, and choose **Properties**.
3. In the displayed dialog box, click **Java Build Path**. On the **Libraries** tab, click **Add JARs** to add the downloaded JAR file.
4. Click **OK**.

Method 2: Adding Maven Dependency

Add the following dependency to the **pom.xml** file to install the Java SDK:

```
<dependency>
<groupId>com.huawei</groupId>
<artifactId>openstack4j</artifactId>
<version>1.0.12</version>
</dependency>
```

For the latest version supported by Java SDK, see **here**.

Method 3: Downloading Java SDK from HUAWEI CLOUD Mirror Center

1. Find the global configuration file **settings.xml** of Maven. Generally, the file is located in **conf** of the Maven installation directory. For example, in Windows, you can locate the configuration file in **D:\maven\apache-maven-3.3.9\conf\settings.xml**.
2. Locate **mirror** in **settings.xml** and add the following configurations:

```
<mirror>
   <id>huaweicloud</id>
   <mirrorOf>*,!HuaweiCloudSDK</mirrorOf>
   <url>https://repo.huaweicloud.com/repository/maven/</url>
</mirror>
```

3. Open the **pom.xml** file of the Maven project and add the following configuration to the **<dependency>** node:

```
<dependency>
   <groupId>com.huawei</groupId>
   <artifactId>openstack4j</artifactId>
  <version>1.0.12</version>
</dependency>
```

For the latest version supported by Java SDK, see **here**.

2.2.2 Java SDK Authentication Modes

Java SDK supports two authentication modes: token-based authentication and AK/SK authentication.

Token Authentication

For details about the code for token-based authentication, see **Table 2-2**.

```
package demo;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import com.huawei.openstack4j.openstack.OSFactory;
import com.huawei.openstack4j.api.OSClient.OSClientV3;
import com.huawei.openstack4j.core.transport.Config;
import com.huawei.openstack4j.model.common.Identifier;
import com.huawei.openstack4j.model.compute.Server;

public class Demo {
    public static void main(String[] args) {
        // Set the authentication parameters.
        String authUrl = "https://iam.example.com/v3";//endpointUrl
        String user = "replace-with-your-username";//username
        String password = "replace-with-your-password";//user password
        String projectId = "replace-with-your-projectId";//project ID
        String userDomainId = "replace-with-your-domainId";//account ID

        // Initialize the client.
        OSClientV3 os = OSFactory.builderV3()
                .endpoint(authUrl)
                .credentials(user, password, Identifier.byId(userDomainId))
                .scopeToProject(Identifier.byId(projectId)).authenticate();

        // Set query parameters.
        Map<String, String> filter = new HashMap<String, String>();

        // Put the parameters that need to be entered into the filter.
        filter.put("limit", "3");

        // Invoke the interface for querying the VM List.
        List<? extends Server> serverList = os.compute().servers().list(filter);
        if (serverList.size() > 0) {
            System.out.println("get serverList success, size = " + serverList.size());
            for (Server server : serverList) {
                System.out.println(server);
            }
        } else {
            System.out.println("no server exists.");
        }
    }
}
```

Table 2-2 Parameter description

Parameter	Description	Example Value
authUrl	Specifies the endpoint of the IAM service. **example** in https://iam.*example*.com/v3 indicates **Region.Cloud platform domain name**. For details about the parameter, see **here**.	https://iam.cn-north-1.myhuaweicloud.com/v3

Parameter	Description	Example Value
user	Specifies the IAM username. For details about how to obtain the username, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
password	Specifies the IAM user password.	N/A
projectId	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	None
userDomainId	Specifies the account ID. For details about how to obtain the account ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	None

The validity period of a token is 24 hours. If your operation time exceeds 24 hours, before you use SDK to invoke APIs, you are advised to reapply the token according to the following method:

```
import com.huawei.openstack4j.openstack.OSFactory;
OSFactory.refreshToken();
```

AK/SK Authentication

For details about the code for AK/SK authentication, see **Table 2-3**.

```
package demo;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import com.huawei.openstack4j.api.OSClient.OSClientAKSK;
import com.huawei.openstack4j.core.transport.Config;
import com.huawei.openstack4j.model.compute.Server;
import com.huawei.openstack4j.openstack.OSFactory;

public class Demo {

 public static void main(String[] args) {

  // Set the authentication parameters.
  String ak = "replace-your-ak";
  String sk = "replace-your-sk";
  String projectId = "replace-your-projectId";
  String region = "replace-your-region"; //example: region = "cn-north-1"
  String cloud = "myhuaweicloud.com";

  OSClientAKSK osclient = OSFactory.builderAKSK().credentials(ak, sk, region, projectId,
```



```
cloud) .authenticate();

 // Set query parameters.
 Map<String , String> filter = new HashMap<String, String>();
 // Put the parameters that need to be entered into the filter.
 filter.put("limit", "3");

 // Invoke the interface for querying the VM List.
 List<? extends Server> serverList = osclient.compute().servers().list(filter);
 if(serverList.size() > 0)
 {
  System.out.println("get serverList success, size = " + serverList.size());
  for (Server server : serverList) {
   System.out.println(server);
  }
 }
 else {
  System.out.println("no server exists.");
 }
 }
}
```

Table 2-3 Parameter description

Parameter	Description	Example Value
ak/sk	Specifies the AK/SK access key. **NOTE** • AK/SK generation description: Log in to the management console, choose **My Credentials**, and click **Access Keys** to create an AK and SK. • The time error between the AK/SK signature time and UTC time cannot exceed 15 minutes. Otherwise, the authentication fails. • If the AK/SK signature fails for more than five consecutive times, the AK/SK request of the source IP address is locked for 5 minutes.	N/A
projectId	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
region	Specifies the region name.	cn-north-1
cloud	Specifies the cloud platform domain name.	myhuaweicloud.com

2.2.3 Java SDK Service Endpoint Configuration

When using SDK to invoke cloud service APIs, you need to obtain the address (endpoint) of each cloud service.

You can use Java SDK to automatically obtain the endpoints or manually encode the endpoints.

The following are examples of manually encoding endpoints for cloud services:

```
endpointResolver.addOverrideEndpoint(ServiceType.CLOUD_EYE, "https://ces.xxx.yyy.com/V1.0/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.DNS, "https://dns.yyy.com");
endpointResolver.addOverrideEndpoint(ServiceType.CDN, "https://cdn.yyy.com/v1.0");
endpointResolver.addOverrideEndpoint(ServiceType.COMPUTE, "https://ecs.xxx.yyy.com/v2/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.NETWORK, "https://vpc.xxx.yyy.com");
endpointResolver.addOverrideEndpoint(ServiceType.IMAGE, "https://ims.xxx.yyy.com");
endpointResolver.addOverrideEndpoint(ServiceType.BLOCK_STORAGE, "https://evs.xxx.yyy.com/v2/%
(project_id)s"");
endpointResolver.addOverrideEndpoint(ServiceType.VOLUME_BACKUP, "https://vbs.xxx.yyy.com/v2/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.AUTO_SCALING, "https://as.xxx.yyy.com/autoscaling-api/
v1/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.LOAD_BALANCER, "https://elb.xxx.yyy.com/v1.0/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.MAP_REDUCE, "https://mrs.xxx.yyy.com/v1.1/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.KEY_MANAGEMENT, "https://kms.xxx.yyy.com/v1.0/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.CLOUD_TRACE, "https://cts.xxx.yyy.com/v1.0/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.ANTI_DDOS, "https://antiddos.xxx.yyy.com/v1/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.Notification, "https://smn.xxx.yyy.com/v2/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.MessageQueue, "https://dms.xxx.yyy.com/v1.0/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.MAAS, "https://maas.xxx.yyy.com/v1/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.ECS, "https://ecs.xxx.yyy.com/v1/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.ECS1_1, "https://ecs.xxx.yyy.com/v1.1/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.EVS, "https://evs.xxx.yyy.com/v2/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.EVS2_1, "https://evs.xxx.yyy.com/v2.1/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.VPC, "https://vpc.xxx.yyy.com/v1/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.VPC2, "https://vpc.xxx.yyy.com/v2.0/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.IDENTITY, "https://iam.xxx.yyy.com/v3");
endpointResolver.addOverrideEndpoint(ServiceType.ORCHESTRATION, "https://rts.xxx.yyy.com/v1/%
(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.NAT, "https://nat.xxx.yyy.com/v2.0");
endpointResolver.addOverrideEndpoint(ServiceType.BMS, "https://bms.xxx.yyy.com/v1/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.DEH, "https://deh.xxx.yyy.com/v1.0/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.CSBS, "https://csbs.xxx.yyy.com/v1/%(project_id)s");
endpointResolver.addOverrideEndpoint(ServiceType.IAM, "https://iam.xxx.yyy.com/v3");
```

- **example** in the preceding code is in **Region.Cloud platform domain name** format. For details about the parameters, see **here**.
- In the preceding code, you do not need to replace the **project_id** value with the actual value.
- Click **here** to obtain a complete code example of using Java SDK for reference.

2.2.4 Java SDK Troubleshooting

Execute the following code to print the execution details of Java SDK:

```
OSFactory.enableHttpLoggingFilter(true);
```


2.3 IAM Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **IAM**.

For details about the parameters supported by SDK, see the **description** of IAM APIs.

The IAM Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
token	**Code**
securitytoken	**Code**
credential	**Code**
region	**Code**
project	**Code (Identity)** **Code (IAM)**
domain	**Code**
user	**Code (Identity)** **Code (IAM)**
group	**Code**
role	**Code**
custom role	**Code**
agency	**Code**
version	**Code**
service	**Code**
endpoint	**Code**

2.4 IMS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of IMS APIs.

The IMS v2 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
image	Code

The Glance v2 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
image	Code
image member	Code

2.5 VPC Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by the SDK when the VPC SDK is used, see the **description** of VPC APIs.

The VPC v1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
bandwidth	Code
port	Code
private ip	Code
public ip	Code
quota	Code
security group	Code
security group rule	Code
subnet	Code
vpc	Code

The VPC v2.0 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
bandwidth	Code
public ip	Code

The Neutron v2.0 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
floating ip	Code
network	Code
port	Code
route	Code
security group	Code
security group rule	Code
subnet	Code

2.6 ECS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of ECS APIs.

The ECS v1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
cloudserver v1	Code
job	Code

The ECS v1.1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
cloudserver	Code

The Nova v2 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
flavor	Code
floating ip	Code
interface	Code
keypair	Code
quota set	Code
security group	Code
server	Code
server group	Code
zone	Code

The following table lists the scenario example code provided by the ECS Java SDK:

Scenario	Example code
create one or more servers	Code
create server with password	Code

2.7 EVS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of EVS APIs.

The EVS v2 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
volume	Code
snapshot	Code
job	Code

The EVS v2.1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
volume	Code

The Cinder v2 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
volume	Code
snapshot	Code
zone	Code
transfer	Code

2.8 AS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **description**.

For details about the parameters supported by SDK, see the **description** of AS APIs.

The AS v1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
group	Code
configration	Code
instance	Code

Resource Object	Example Code
lifecyclehook	Code
log	Code
notification	Code
policy	Code
quota	Code
tag	Code

2.9 Cloud Eye Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of Cloud Eye APIs.

The Cloud Eye V1.0 Java SDK supports the following resource. The table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
metric	Code
alarm	Code
metric data	Code
quota	Code

2.10 DNS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **DNS**.

For details about the parameters supported by SDK, see the **description** of DNS APIs.

The Domain Name Service (DNS) v2 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
zone	Code
recordset	Code

Resource Object	Example Code
ptr	Code

2.11 ELB Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of ELB APIs.

The ELB (Enhanced Load Balancer) v2 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
loadbalancer	Code
listener	Code
pool	Code
member	Code
healthmonitor	Code
certificate	Code
whitelist	Code
l7policy	Code
l7rule	Code

The CLB (Classic Load Balancer) v1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
loadbalancer	Code
listener	Code
member	Code
healthcheck	Code
certificate	Code

2.12 VBS Java SDK Demo

Creating a VBS Backup

You can create a VBS backup using OpenStack4j based on the following code. After the VBS backup is created, it will be displayed in the VBS list on the VBS console.

```
public static void createBackup() {
    AsyncVolumeBackupCreate vbc = Builders.asyncVolumeBackupCreate()
            .name(backupName)
            .volumeId(volume.getId())
            .build();
    AsyncVolumeBackupJob job = osclient.blockStorage().asyncBackups().create(vbc);
    Assert.assertNotNull(job.getId());
    backupJobId = job.getId();
}
```

Table 2-4 Request parameter description

Parameter	Mandatory	Type	Description
backup	Yes	dict	Specifies the backup to be created.
volume_id	Yes	string	Specifies the ID of the disk to be backed up.
snapshot_id	No	string	Specifies the snapshot ID of the disk to be backed up.
name	Yes	string	Specifies the backup name. The value is a string of 1 to 64 characters consisting of digits, letters, underscores (_), and hyphens (-).
description	No	string	Provides supplementary information about the backup. The value is a string of 1 to 64 characters and cannot contain the less-than sign (<) or greater-than sign (>).

Querying VBS Backup Details

You can query the backup list and obtain the backup details using OpenStack4j based on the following code:

```
public static void queryNativeBackupsDetail(){
    // Without specifying the search criteria
    List<? extends VolumeBackup> list = osclient.blockStorage().backups().list(true);
    Assert.assertNotEquals(list.size(), 0);

    // With the search criteria specified
    HashMap<String, String> filter = new HashMap<>();
    filter.put("name", backupName);
```



```
    List<? extends VolumeBackup> list2 = osclient.blockStorage().backups().list(true, filter);
    for (VolumeBackup backup: list2) {
        Assert.assertEquals(backup.getName(), backupName);
    }
}
```

Table 2-5 Request parameter description

Parameter	Mandatory	Type	Description
name	No	string	Specifies the name of the backup to be queried. This parameter is used to query the backups whose names are specified character strings.
status	No	string	Specifies the status of the backup to be queried. This parameter is used to query the backups in a specified state. The value can be **available**, **error**, **restoring**, **creating**, **deleting**, or **error_deleting**.
offset	No	int	Specifies the offset of the queried details.
limit	No	int	Specifies the maximum number of query results that can be returned.
volume_id	No	string	Specifies the disk ID of the backup to be queried. This parameter is used to query the backups for specific disks.

Restoring a Disk Using a VBS Backup

You can restore a disk from a VBS backup using OpenStack4j based on the following code:

```
public static void restoreBackup() {
    AsyncVolumeBackupJob job = osclient.blockStorage()
            .asyncBackups()
            .restore(backupId, volume.getId());
    Assert.assertNotNull(job.getId());
}
```

Table 2-6 Request parameter description

Parameter	Mandatory	Type	Description
restore	Yes	dict	Specifies the operation of restoring the disk using a backup.
backup_id	Yes	string	Specifies the ID of the backup used to restore a disk.
volume_id	Yes	string	Specifies the ID of the disk to be restored.

Deleting a Backup

You can delete a backup using OpenStack4j based on the following code:

```
public static void deleteNativeBackup()
{
    ActionResponse delete = osclient.blockStorage().backups().delete(backupId);
    Assert.assertEquals(delete.isSuccess(), true);
}
```

Table 2-7 Request parameter description

Parameter	Mandatory	Type	Description
tenant_id	Yes	string	Specifies the ID of the tenant.
backup_id	Yes	string	Specifies the ID of the backup used to restore a disk.

Creating a Backup Policy

You can create a backup policy using OpenStack4j based on the following code:

```
public static void createPolicy()
{
    // Create a scheduled policy first.
    VBSVolumeBackupScheduledPolicy scheduledPolicy = VBSVolumeBackupScheduledPolicy.builder()
            .frequency(10)
            .maxBackupAmount(10)
            .retainFirstBackupOfCurrentMonth(true)
            .startTime("01:00")
            .status(VolumeBackupPolicy.VolumeBackupPolicyStatus.OFF)
            .build();
    Assert.assertNotNull(scheduledPolicy);

    // Create a backup policy object.
    VolumeBackupPolicy create = VBSVolumeBackupPolicy.builder()
            .name(policyName)
            .scheduledPolicy(scheduledPolicy)
            .build();
    VolumeBackupPolicy policy = osclient.blockStorage().policies().create(create);
    Assert.assertNotNull(policy.getId());
```

Table 2-8 Request parameter description

Parameter	Mandatory	Type	Description
backup_policy_name	Yes	string	Specifies the backup policy name. The name is a string of 1 to 64 characters consisting of letters, digits, underscores (_), and hyphens (-). It cannot start with **default**.
scheduled_policy	Yes	dict	Specifies details about the scheduling policy.

Parameter	Mandatory	Type	Description
start_time	Yes	string	Specifies the backup start time, which needs to be converted into the local UTC time (on the hour only). The value is in **HH:mm** format.
frequency	No	integer	Specifies the backup interval (1 to 14 days). Select either this parameter or **week_frequency**. If you select both, this parameter is used.
week_frequency	No	list<dict>	Specifies on which days of each week backup jobs are executed. The value can be one or more of the following: SUN, MON, TUE, WED, THU, FRI, SAT
rentention_num	No	integer	Specifies the retained number (minimum: 2) of backups. Select either this parameter or **rentention_day**. If you select both, this parameter is used.
rentention_day	No	integer	Specifies how many days backups are retained.
remain_first_backup_of_curMonth	Yes	string	Specifies whether to retain the first backup in the current month. The value can be **Y** or **N**.
status	Yes	string	Specifies the backup policy status. The value can be **ON** or **OFF**.

Deleting a Backup Policy

You can delete a backup policy using OpenStack4j based on the following code:

```
public static void deletePolicy() {
    osclient.blockStorage().policies().delete(policyId);
    List<? extends VolumeBackupPolicy> policies = osclient.blockStorage().policies().list();
    boolean isSuccess = true;
    for (VolumeBackupPolicy policy:
            policies) {
        if (policy.getId().equals(policyId)) {
            isSuccess = false;
            break;
        }
    }
    Assert.assertEquals(isSuccess, true);
}
```


Table 2-9 Request parameter description

Parameter	Mandatory	Type	Description
tenant_id	Yes	string	Specifies the ID of the tenant.
policy_id	Yes	string	Specifies the ID of the policy.

Querying Backup Policies

You can query backup policies using OpenStack4j based on the following code:

```
public static void queryPolicy() {
    List<? extends VolumeBackupPolicy> policies = osclient.blockStorage().policies().list();
    boolean isSuccess = false;
    for (VolumeBackupPolicy policy:
         policies) {
        if (policy.getName().equals(policyName)) {
            isSuccess = true;
            policyId = policy.getId();
            break;
        }
    }
    Assert.assertEquals(isSuccess, true);
}
```

2.13 CTS Java SDK Demo

Tracker

A tracker will be created after CTS is enabled. All traces recorded by CTS are associated with the tracker.

Creating a Tracker

You can create a tracker using OpenStack4j based on the following code. Currently, only one tracker **system** can be created.

```
public void CreateTracker() {
Tracker create = osclient.cloudTraceV1().trackers().create(bucket_name,FilePrefixName);
}
```

Deleting a Tracker

You can delete a tracker using OpenStack4j based on the following code:

```
public void DeleteTraker() {
ActionResponse delete = osclient.cloudTraceV1().trackers().delete(tracker_name);
Assert.assertTrue(delete.isSuccess());
List<Tracker> trackers = osclient.cloudTraceV1().trackers().list();
Assert.assertTrue(trackers.size() == 0);
}
```


Updating a Tracker

You can update a tracker using OpenStack4j based on the following code. The information that can be updated includes the bucket name, folder name, status, and the tracker name. The tracker name is optional and can be only **system**.

```
public void UpdateTraker() {
TrackerUpdate update = TrackerUpdate.builder().trackerName(tracker_name)
.bucketName(bucket_name).filePrefixName("SDK-unittest").status(TrackerStatus.Enabled").build();
Tracker updated = osclient.cloudTraceV1().trackers().update(update);
}
```

Querying a Tracker

You can query a tracker using OpenStack4j based on the following code by specifying **tracker_name**:

```
public void GetTraker() {
Tracker get = osclient.cloudTraceV1().trackers().get(tracker_name);
}
```

Trace

This interface is used to query records of operations on resources during the last seven days.

Querying the Trace List

You can query a trace list using OpenStack4j based on the following code. You can filter out required traces by specifying multiple parameters.

```
/*v2interface*/
publicvoidListTrace(){
    TraceListOptionsoptions=TraceListOptions.create().limit(5).user("renxiaomei").serviceType("CTS");
    List<Trace>list=osclient.cloudTraceV2().traces().list("system", options);
    if(list.size()>0){
        Tracetrace=list.get(list.size()-1);
        options.marker(trace.getId());
        List<Trace>list2=osclient.cloudTraceV2().traces().list("system", options);
    }
}
```

2.14 KMS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of KMS APIs.

The following table lists the scenario example code provided by the KMS Java SDK:

Scenario	Example Code
manage customer master key	**Code**
encrypt/decrypt data	**Code**

Scenario	Example Code
create data encrypt key	Code
encrypt/decrypt with data encrypt key	Code

2.15 Anti-DDoS Java SDK Demo

Querying Optional Anti-DDoS Defense Policies

You can query optional Anti-DDoS defense policies. Based on your service, you can select a policy for Anti-DDoS traffic cleaning.

Code example:

```
public void listConfigs()
   {
       AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
       AntiDDoSService antiDDoSService=antiDDoSServices.antiddos();
       AntiDDoSConfig configs = antiDDoSService.listConfigs();
       LOGGER.info("{}", configs);
   }
```

Enabling Anti-DDoS

You can enable Anti-DDoS traffic cleaning defense. Successfully invoking this API only means that the service node has received the enabling request. You need to use the task querying API to check the task execution status.

Code example:

```
public void createAntiDDoS() throws InterruptedException
   {
       AntiDDoS entity = AntiDDoS.builder()
           .enableL7(true)
           .trafficPos(TrafficPos.POS_1)
           .httpRequestPos(HttpRequestPos.POS_1)
           .cleaningAccessPos(CleaningAccessPos.POS_1)
           .appType(AppType.Type_0)
           .build();
       AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
       AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
       Task task = antiDDoSService.create(entity, floatingIpId);
       LOGGER.info("{}", task);
       taskId = task.getTaskId();
       waitTaskFinish(taskId);    }
   }
}
```

Disabling Anti-DDoS

You can disable Anti-DDoS traffic cleaning defense. Successfully invoking this API only means that the service node has received the disabling request. You need to use the task querying API to check the task execution status.

Code example:

```
public void deleteAntiDDoS()
       throws InterruptedException
```



```
    {
        AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
        AntiDDoSService antiDDoSService=antiDDoSServices.antiddos();
        Task task = antiDDoSService.delete(floatingIpId);
        LOGGER.info("{}", task);
        waitTaskFinish(task.getTaskId());
}
```

Querying Configured Anti-DDoS Defense Policies

You can query the configured Anti-DDoS defense policy of a specified EIP.

Code example:

```
public void getAntiDDoS()
    {
        AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
        AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
        AntiDDoS antiDDoS = antiDDoSService.get(floatingIpId);
        LOGGER.info("{}", antiDDoS);
    }
```

Updating Anti-DDoS Defense Policies

You can update the Anti-DDoS defense policy of a specified EIP. Successfully invoking this API only means that the service node has received the update request. You need to use the task querying API to check the task execution status.

Code example:

```
public void updateAntiDDoS() throws InterruptedException
    {
        AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
        AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
        AntiDDoS entity = antiDDoSService.get(floatingIpId);
        entity = entity.toBuilder().appType(AppType.Type_1).build();
        Task task = osclient.antiDDoS().antiddos().update(entity, floatingIpId);
        LOGGER.info("{}", task);
        waitTaskFinish(task.getTaskId());
    }
```

Querying Anti-DDoS Tasks

You can query the execution status of a specified Anti-DDoS configuration task.

Code example:

```
public void getTask()
    {
        AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
        AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
        Task task = antiDDoSService.getTask(taskId);
        LOGGER.info("{}", task);
}
```

Querying the List of EIP Defense Statuses

You can query the defense statuses of all EIPs, regardless whether an EIP has been bound to an ECS or not.

Code example:

```
public void listStatuses()
{
```



```
        AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
        AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
        AntiDDoSStatus statuses = antiDDoSService.listStatus();
        LOGGER.info("{}", statuses);
    }
```

Querying the Defense Status of a Specified EIP

You can query the defense status of a specified EIP.

Code example:

```
public void getStatus()
    {
        AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
        AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
        AntiDDoSStatusDetail status = antiDDoSService.getStatus(floatingIpId);
        LOGGER.info("{}", status);
    }
```

Querying the Traffic of a Specified EIP

You can query the traffic of a specified EIP in the last 24 hours. Traffic is detected at five-minute intervals.

Code example:

```
public void dailyReport()
    {
        AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
        AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
        List<? extends AntiDDoSDailyData> dailyReport = antiDDoSService.dailyReport(floatingIpId);
        LOGGER.info("{}", dailyReport);
    }
```

Querying Events of a Specified EIP

You can query events of a specified EIP in the last 24 hours. Events include cleaning and blackhole events, and the query delay is within five minutes.

Code example:

```
public void listLog()
    {
        AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
        AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
        List<? extends AntiDDoSLog> logs = antiDDoSService.listLogs(floatingIpId);
        LOGGER.info("{}", logs);

        AntiDDoSLogListOptions options = AntiDDoSLogListOptions.create().limit(1).offset(1);
        List<? extends AntiDDoSLog> logs2 = osclient.antiDDoS().antiddos().listLogs(floatingIpId, options);
        LOGGER.info("{}", logs2);
    }
```

Querying Weekly Defense Statistics

You can query weekly defense statistics about all your EIPs, including the number of intercepted DDoS attacks, number of attacks, and ranking by the number of attacks.

Code example:

```
public void weeklyReport()
    {
```



```
    AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
    AntiDDoSService antiDDoSService =antiDDoSServices.antiddos();
    AntiDDoSWeeklyData weekly = antiDDoSService.weeklyReport();
    LOGGER.info("{}", weekly);

    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.HOUR, -10);
    AntiDDoSWeeklyData weekly2 = osclient.antiDDoS().antiddos().weeklyReport(cal.getTime());
    LOGGER.info("{}", weekly2);
  }
```

Querying Alarm Configuration

You can query alarm configuration, such as whether a certain type of alarms will be received, and whether alarms are received through SMS messages or emails.

Code example:

```
public void queryWarningInfo()
  {
    AntiDDoSServices antiDDoSServices = osclient.antiDDoS();
    AntiDDoSWarn query = antiDDoSServices.warnalert().query();
    LOGGER.info("{}", query);
}
```

2.16 DMS Java SDK Demo

Distributed Message Service (DMS) is a message middleware service based on distributed, high-availability clustering technology. It provides reliable, scalable, fully managed queues for storing messages. DMS enables cloud applications to decouple from each other, achieving high cost-effectiveness.

Creating a Queue

You can create a queue using OpenStack4j based on the following code. After the queue is created, messages will be sent to this queue.

```
String name = randomName();
String description = "sdk-unittest"
Queue queue = null;
queue = osclient.messageQueue().queue().create(name, description);
```

Creating a Consumer Group

You can create a consumer group using OpenStack4j based on the following code. After the consumer group is created, it can consume messages in the queue.

```
List<ConsumerGroup> groups = null;
List<String> groupNames = Lists.newArrayList("consumer-group-1", "consumer-group-2");
queueId queueID = queue.getId();
groups = osclient.messageQueue().consumerGroups().create(queueID, groupNames);
```

Producing Messages

You can produce messages using OpenStack4j based on the following code:

```
public void testProduceMessage() {
HashMap<String, Object> attributes1 = Maps.newHashMap();
attributes1.put("attr1", 1);
attributes1.put("attr2", false);
QueueMessage message = QueueMessage.builder().body("sdk-unittests").attributes(attributes1).build();
```



```
ActionResponse produce = osclient.messageQueue().messages().produce(queue.getId(), message);
}
```

Consume Messages

You can consume messages using OpenStack4j based on the following code:

```
public void testConsumeMessages() {
ConsumerGroup consumerGroup1 = groups.get(0);
List<QueueMessageWithHandler> all = Lists.newArrayList();
for (int i = 0; i < 3; i++) {
List<QueueMessageWithHandler> temp = osclient.messageQueue().messages().consume(queue.getId(),
consumerGroup1.getId(), 5, 10);
all.addAll(temp);
}
}
```

2.17 MRS Java SDK Demo

MapReduce Service (MRS) provides users with a complete controllable, enterprise-class, big data cloud service, ensuring the smooth running of big data components, such as Hadoop, Spark, HBase, Kafka, and Storm.

Creating a Cluster and Submitting the Job

You can create a cluster and submit the job using OpenStack4j based on the following code. After the cluster is created, it will be displayed on the cluster page of the MRS console.

```
public void createClusterAndRunAJob()
{
MapReduceComponent component =
MapReduceComponent.builder().id(component_id).name(component_name).version(component_version).de
sc(component_desc).build();
MapReduceClusterCreate cluster =
MapReduceClusterCreate.builder().dataCenter(data_center).masterNodeNum(master_node_num).masterNo
deSize(master_node_size).coreNodeNum(core_node_num).coreNodeSize(core_node_size).name(cluster_nam
e).availablilityZoneId(available_zone_id).vpcName(vpc).vpcId(vpc_id).subnetName(subnet_name).subnetId(s
ubnet_id).version(cluster_version).type(cluster_type).volumeSize(volume_size).volumeType(volume_type).key
pair(node_public_cert_name).safeMode(safe_mode).components(Lists.newArrayList(component)).build();
MapReduceJobExeCreate jobExe =
MapReduceJobExeCreate.builder().jobType(job_type).jobName(job_name).jarPath(jar_path).arguments(argu
ments).input(input).output(output).jobLog(job_log).fileAction(file_action).hql(hql).hiveScriptPath(hive_script
_path).shutdownCluster(shutdown_cluster).submitJobOnceClusterRun(submit_job_once_cluster_run).build();
MapReduceClusterCreateResult result = osclient.mrs().clusters().createAndRunJob(cluster, jobExe);
}
```

Querying Cluster Details

You can query details of a cluster using OpenStack4j based on the following code by specifying the cluster ID:

```
public void describeCluster () {
MapReduceClusterInfo cluster = osclient.mrs().clusters().get(id);
}
```

Terminating a Cluster

You can terminate a cluster using OpenStack4j based on the following code by specifying the cluster ID:


```
public void deleteCluster () {
ActionResponse delete = osclient.mrs().clusters().delete(id);
}
```

Adding and Executing a Job

You can add a job and execute using OpenStack4j based on the following code. After the job is created, it will be displayed on the job page of the MRS console.

```
public void submitAndExecuteJob () {
MapReduceJobExeCreate jobExeCreate =
MapReduceJobExeCreate.builder().jobType(job_type).jobName(job_name).clusterId(cluster_id).jarPath(jar_pa
th).arguments(arguments).input(input).output(output).jobLog(job_log).fileAction(file_action).hql(hql).hiveSc
riptPath(hive_script_path).isProtected(is_protected).isPublic(is_public).build();
MapReduceJobExe exe = osclient.mrs().jobExes().create(jobExeCreate);
}
```

Querying the Job Exe Object List

You can query job exe object list using OpenStack4j based on the following code:

```
public void getJobExeList () {
JobExeListOptions options =
JobExeListOptions.create().page(current_page).pageSize(page_size).clusterId(cluster_id)
.state(state);
List<? extends MapReduceJobExe> list = osclient.mrs().jobExes().list(options);
}
```

Querying Details of a Job Exe Object

You can query details of a job exe object using OpenStack4j based on the following code by specifying the object ID:

```
public void getJobExes() {
osclient.mrs().jobExes().get(id);
}
```

Deleting a Job Execution Object

You can delete a job execution object using OpenStack4j based on the following code by specifying the object ID:

```
public void deleteJobExecution () {
ActionResponse delete = osclient.mrs().jobExecutions().delete(id);
}
```

2.18 CDN Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of CDN APIs.

The CDN Java SDK supports operations on the following resource objects.

Resource Object	Example Code
Domain	**Code**

Resource Object	Example Code
Statistic	Code
Log	Code

2.19 FGS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **FGS**.

For details about the parameters supported by SDK, see the **description** of FGS APIs.

The FGS V1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
function	Code
version	Code
trigger	Code

The FGS V2 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
function	Code
version	Code
trigger	Code

2.20 TMS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by TMS SDK, see the **description**.

The TMS v1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
Tag management	Code

2.21 EPS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by EPS SDK, see the **description**.

The EPS v1 Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
EnterpriseProject management	Code

2.22 RDS Java SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by RDS SDK, see **description** of RDS APIs.

The RDS Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
Rds Instance management	Code

The following table lists the code provided by SDK.

Scenario	Example Code
Rds Backup and Restore	Code
Rds Param Configurtion	Code

3 Python

3.1 Getting Started with Python SDK

Welcome to use HUAWEI CLOUD developer tools (Python SDK). Python SDK allows you to easily access cloud services using codes.

This tutorial describes how to install and use Python SDK and provides examples to help you quickly get started.

The supported Python SDK is developed based on the Python OpenStack SDK.

Supported Cloud Services

The Python SDK supports the following cloud services:

- **Identity and Access Management (IAM)**
- **Image Management Service (IMS)**
- **Virtual Private Cloud (VPC)**
- **Elastic Cloud Server (ECS)**
- **Elastic Volume Service (EVS)**
- **Auto Scaling (AS)**
- **Cloud Eye (CES)**
- **Domain Name Service (DNS)**
- **Elastic Load Balancing (ELB)**
- **Volume Backup Service (VBS)**
- **Key Management System (KMS)**
- **Anti-DDoS**
- **Distributed Message Service (DMS)**
- **MapReduce Service (MRS)**
- **Relational Database Service (RDS)**
- **Content Delivery Network (CDN)**
- **Tag Management Service (TMS)**

- **Enterprise Management (EPS)**

Prerequisites

1. You have obtained a cloud platform account and provisioned all required services.
2. You have installed Python, pip, and git. Python SDK is applicable to Python 2.7.10 to 2.7.15 and 3.4 to 3.7.

SDK Acquisition and Installation

If the pip is used for installation, run the following command:

pip install huaweicloud-sdk-python

How to Use

Set parameters, create connections, and invoke the SDK to access the service API. For details about the parameters, see **Table 3-1**.

```
# -*- coding:utf-8 -*-
from openstack import connection

# create connection
username = "replace-with-your-username" # username
password = "replace-with-your-password" # user password
projectId = "replace-with-your-projectId"    # project ID
userDomainId = "replace-with-your-domainId"  # account ID
auth_url = "https://iam.example.com/v3"    # endpoint url
conn = connection.Connection(auth_url=auth_url,
                             user_domain_id=userDomainId,
                             project_id=projectId,
                             username=username,
                             password=password)

# set parameters
limit = 5

# define function for listing servers
def list_servers():
    # get server list with params
    servers = conn.compute.servers(limit=limit)
    # iterate servers list
    for server in servers:
        print(server)

# visit API
list_servers()
```


Table 3-1 Parameter description

Parameter	Description	Example Value
auth_url	Specifies the endpoint of the IAM service. **example** in the https://iam.*example*.com/v3, indicates the **Region.Cloud platform domain name**. For details about the parameter, see **here**.	https://iam.cn-north-1.myhuaweicloud.com/v3
username	Specifies the IAM username. For details about how to obtain the username, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
password	Specifies the IAM user password.	N/A
projectId	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
userDomainId	Specifies the account ID. For details about how to obtain the account ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A

3.2 Using the Python SDK

3.2.1 Installing the Python SDK

The Python SDK provides GitHub and pip two installation modes.

Method 1: Downloading the GitHub Source Code

Download the code to a proper position of your project. Use *pythonsdk* as an example. Run the following commands to download the source code and install the SDK:

git clone https://github.com/huaweicloud/huaweicloud-sdk-python *pythonsdk*

cd *pythonsdk*

pip install -r requirements.txt

python setup.py install

Method 2: Run the Following Command:

pip install huaweicloud-sdk-python

3.2.2 Python SDK Authentication Modes

Python SDK supports two authentication modes: token-based authentication and AK/SK authentication.

Token Authentication

For details about the code for token-based authentication, see **Table 3-2**.

```
# -*- coding:utf-8 -*-
from openstack import connection

# create connection
username = "replace-with-your-username" # username
password = "replace-with-your-password" # user password
projectId = "replace-with-your-projectId"    # project ID
userDomainId = "replace-with-your-domainId"  # account ID
auth_url = "https://iam.example.com/v3"    # endpoint url
conn = connection.Connection(auth_url=auth_url,
                             user_domain_id=userDomainId,
                             project_id=projectId,
                             username=username,
                             password=password)

# set parameters
limit = 5

# define function for listing servers
def list_servers():
    # get server list with params
    servers = conn.compute.servers(limit=limit)
    # iterate servers list
    for server in servers:
        print(server)

# visit API
list_servers()
```

Table 3-2 Parameter description

Parameter	Description	Example Value
auth_url	Specifies the endpoint of the IAM service. **example** in the https://iam.*example*.com/v3, indicates the **Region.Cloud platform domain name**. For details about the parameter, see **here**.	https://iam.cn-north-1.myhuaweicloud.com/v3

Parameter	Description	Example Value
username	Specifies the IAM username. For details about how to obtain the username, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
password	Specifies the IAM user password.	N/A
projectId	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
userDomainId	Specifies the account ID. For details about how to obtain the account ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A

AK/SK Authentication

For details about the code for AK/SK authentication, see **Table 3-3**.

```
#encoding=utf-8

from openstack import connection

projectId = "***"
cloud = "myhuaweicloud.com"
region= "***"     # example: region = "cn-north-1"
AK = "***"
SK = "***"

conn = connection.Connection(
              project_id=projectId,
              cloud=cloud,
              region=region,
              ak = AK,
              sk = SK)

def test_compute():
    servers = conn.compute.servers(limit = 3)
    for server in servers:
        print server

if __name__ == "__main__":
    test_compute()
```


Table 3-3 Parameter description

Parameter	Description	Example Value
ak/sk	Specifies the AK/SK access key. **NOTE** • AK/SK generation description: Log in to the management console, choose **My Credentials**, and click **Access Keys** to create an AK and SK. • The time error between the AK/SK signature time and UTC time cannot exceed 15 minutes. Otherwise, the authentication fails. • If the AK/SK signature fails for more than five consecutive times, the AK/SK request of the source IP address is locked for 5 minutes.	N/A
project_id	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
region	Specifies the region name.	cn-north-1
cloud	Specifies the cloud platform domain name.	myhuaweicloud.com

3.2.3 Python SDK Configuration on the Client

Some functions supported by SDK can be enabled or disabled through configuration.

Sample code

```
conn = connection.Connection(auth_url=auth_url,
                             user_domain_id=userDomainId,
                             project_id=projectId,
                             username=username,
                             password=password,
                             verify=False)
```

Currently, the following custom parameters are supported.

Parameter	Default Value	Function Description	Remarks
verify	True	SSL check	You are advised to set **verify** to **True**.

3.2.4 Python SDK Service Endpoint Configuration

When using SDK to invoke cloud service APIs, you need to obtain the address (endpoint) of each cloud service.

You can use Python SDK to automatically obtain the endpoints or manually encode the endpoints.

The following are examples of manually encoding endpoints for cloud services:

```
os.environ.setdefault(
    'OS_CLOUD_EYE_ENDPOINT_OVERRIDE',
    'https://ces.example.com/V1.0/%(project_id)s'
)
os.environ.setdefault(
    'OS_AUTO_SCALING_ENDPOINT_OVERRIDE',
    ('https://as.example.com'
     '/autoscaling-api/v1/%(project_id)s')
)
os.environ.setdefault(
    'OS_DNS_ENDPOINT_OVERRIDE',
    'https://dns.example.com/v2'
)
os.environ.setdefault(
    'OS_VOLUME_BACKUP_ENDPOINT_OVERRIDE',
    'https://vbs.example.com/v2/%(project_id)s'
)
os.environ.setdefault(
    'OS_ELBV1_ENDPOINT_OVERRIDE',
    'https://elb.example.com/v1.0/%(project_id)s'
)
os.environ.setdefault(
    'OS_MAP_REDUCE_ENDPOINT_OVERRIDE',
    'https://mrs.example.com/v1.1/%(project_id)s'
)
os.environ.setdefault(
    'OS_CTS_ENDPOINT_OVERRIDE',
    'https://cts.example.com/v1.0/%(project_id)s'
)
os.environ.setdefault(
    'OS_SMN_ENDPOINT_OVERRIDE',
    'https://smn.example.com/v2/%(project_id)s'
)
os.environ.setdefault(
    'OS_MAAS_ENDPOINT_OVERRIDE',
    'https://maas.example.com/v1/%(project_id)s'
)
os.environ.setdefault(
    'OS_KMS_ENDPOINT_OVERRIDE',
    'https://kms.example.com/v1.0/%(project_id)s'
)
os.environ.setdefault(
    'OS_ANTI_DDOS_ENDPOINT_OVERRIDE',
    'https://antiddos.example.com/v1/%(project_id)s'
)
os.environ.setdefault(
    'OS_DMS_ENDPOINT_OVERRIDE',
```



```
    'https://dms.example.com/v1.0/%(project_id)s'
)
os.environ.setdefault(
    'OS_RDSV1_ENDPOINT_OVERRIDE',
    'https://rds.example.com/v3/%(project_id)s'
)
os.environ.setdefault(
    'OS_CDN_ENDPOINT_OVERRIDE',
    'https://cdn.example.com/v1.0'
)
```

- **example** in the preceding code is in **Region.Cloud platform domain name** format. For details about the parameter, see **here**.
- In the preceding code, you do not need to replace the **project_id** value with the actual value.
- Click **here** to obtain a complete code example of using Python SDK for reference.

3.2.5 Python SDK Troubleshooting

To enable the debugging function using Python SDK, add the following code to the application:

```
from openstack import utils
utils.enable_logging(debug=True,stream=sys.stdout)
```

3.2.6 Common Parameters of Python SDK

The Python SDK supports some common parameters. They are used as follows:

ignore_missing

This parameter is usually displayed in the delete interface of a service. The default value is **True**. For example:

delete_flavor(self, flavor, ignore_missing=True)

delete_image(self, image, ignore_missing=True)

If this parameter is set to **True** and the deleted resource does not exist, no exception information is displayed and thrown.

If this parameter is set to **False** and the deleted resource does not exist, the **openstack.exceptions.ResourceNotFound** exception will be thrown. If the exception is not handled, the program stops abnormally.

details

This parameter is usually displayed in the interface for obtaining the detailed resource list. For example:

flavors(self, details=True, **query)

images(self, details=True, **query)

Only the resources that support /detail interface can be imported.

If this parameter is set to **True** and the URI of the requested resource is **{RES} / details**, return the detailed information of the resource.

If this parameter is set to **False** and the URI of the requested resource is **{RES}**, return the most basic information of the resource.

limit

This parameter is usually displayed in the interface for obtaining the detailed resource list. For example:

flavors(self, details=True, limit = 5)

images(self, details=True, limit = 10)

When the **limit** parameter is imported, the returned results are displayed in pages. The number of returned results equates the **limit** value.

If the **limit** parameter is not imported, the query results are returned once.

3.3 IAM Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **IAM**.

For details about the parameters supported by SDK, see the **description** IAM API.

The IAM Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
authtoken	**Code**
securitytoken	**Code**
credential	**Code**
region	**Code**
project	**Code (Identity)** **Code (IAM)**
domain	**Code**
user	**Code (Identity)** **Code (IAM)**
group	**Code**
role	**Code**
customrole	**Code**
agency	**Code**
version	**Code**
service	**Code**

Resource Object	Example Code
endpoint	Code

3.4 IMS Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of IMS APIs.

The IMS v2 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
image	Code

The Glance v2 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
image	Code
image member	Code

3.5 VPC Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by the SDK when the VPC SDK is used, see the **description** of VPC APIs.

The VPC v1 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
bandwidth	Code
port	Code

Resource Object	Example Code
private ip	Code
public ip	Code
quota	Code
security group	Code
security group rule	Code
subnet	Code
vpc	Code

The VPC v2.0 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
sharebandwidth	Code

The Neutron v2.0 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
floating ip	Code
network	Code
port	Code
route	Code
security group	Code
security group rule	Code
subnet	Code

3.6 ECS Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of ECS APIs.

The ECS Java SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
availability zone	Code
flavor	Code
job	Code
keypair	Code
server (v2)	Code
server group	Code
server interface	Code
server ip	Code
server tag	Code
cloudserver (v1.1)	Code
cloudserver (v1)	Code

The following table lists the scenario example codes provided by the ECS Java SDK:

Scenario	Example Code
create one or more servers	Code
create server with password	Code

3.7 EVS Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **EVS**.

For details about the parameters supported by SDK, see the **description** of EVS APIs.

The EVS v2 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
volume	Code

The Cinder v2 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
volume	Code
snapshot	Code
type	Code

3.8 AS Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **AS**.

For details about the parameters supported by SDK, see the **description** of AS APIs.

The AS v1 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
group	Code
config	Code
hook	Code
instance	Code
policy	Code
quota	Code
activity	Code

3.9 Cloud Eye Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of Cloud Eye APIs.

The Cloud Eye V1.0 Python SDK supports the following resource. The table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
metric	Code
alarm	Code
metric data	Code
quota	Code

3.10 DNS Python SDK Demo

Service Description

Domain Name Service (DNS) provides highly available and scalable authoritative DNS resolution services and domain name management services. It translates domain names or application resources into IP addresses required for network connection. By doing so, visitors' access requests are directed to the desired resources.

Creating a Private Zone

To use the DNS service to manage domain names in VPCs, you need to configure private zones on the DNS console. You can use the Python OpenStack SDK to create a private zone as follows:

1. Specify the VPC to be associated.
2. Create a private zone.

You can create a private zone using the Python OpenStack SDK based on the following code. After the private zone is created, it will be displayed on the private zone page of the DNS console.

```
def setUpClass(cls):
    super(TestZone, cls).setUpClass()
    # get a router
    routers = cls.conn.network.routers(limit=2)
    idx = 0
    for _router in routers:
        idx += 1
        print _router
        if idx == 1:
            cls.router = _router
        if idx == 2:
            cls.router2 = _router
            break
    # create zone
    cls.zone = auto_create_private_zone(cls.conn, cls.NAME, cls.router.id,region)
```

Associating a VPC

You can use the Python OpenStack SDK to associate a private zone with a VPC on the cloud platform. The association procedure is as follows:

1. Specify the VPC to be associated.

2. Select the target private zone to associate with the VPC.

You can associate a private zone with a VPC using the Python OpenStack SDK based on the following code:

```
def add_router_to_zone(self):
    # Designate a router
    resource2.wait_for_status(self.conn.dns._session, self.zone, "ACTIVE", interval=5, failures=["ERROR"])
    # Associate the private zone to the router
    result = self.conn.dns.add_router_to_zone(self.zone, **{"router_id": self.router2.id,"router_region":
region})
    self.assertEqual(result.router_id, self.router2.id)
    self.assertEqual(result.router_region, region)
    zone = self.conn.dns.get_zone(self.zone)
    self.assertEqual(2, len(zone.routers))
    router_ids = [_router["router_id"] for _router in zone.routers]      self.assertIn(self.router.id, router_ids)
```

Disassociating a VPC

You can use the Python OpenStack SDK to disassociate a private zone from a VPC on the cloud platform. The code is as follows:

```
def remove_router_of_zone(self):
    resource2.wait_for_status(self.conn.dns._session, self.zone, "ACTIVE", interval=5, failures=["ERROR"])
    result = self.conn.dns.remove_router_from_zone(self.zone, **{          "router_id": self.router.id,
      "router_region": region
    })
    self.assertEqual(result.router_id, self.router.id)
    self.assertEqual(result.router_region, region)
```

Deleting a Private Zone

You can delete a private zone that you do not need to manage using the DNS service. After the deletion, domain names included in this zone cannot be resolved.

Before deleting a private zone, ensure that all record sets in this zone have been backed up. The code is as follows:

```
def tearDownClass(cls):
    # delete zone
    cls.conn.dns.delete_zone(cls.zone)
```

3.11 ELB Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of ELB APIs.

The ELB (Enhanced Load Balancer) v2 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
loadbalacner	**Code**
listener	**Code**

Resource Object	Example Code
pool	Code
member	Code
healthmonitor	Code
certificate	Code
l7policy	Code
l7rule	Code
whitelist	Code

The CLB (Classic Load Balancer) v1 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
loadbalacner	Code
listener	Code
member	Code
healthcheck	Code
certificate	Code

3.12 VBS Python SDK Demo

Creating a VBS Backup

You can create a VBS backup using the Python OpenStack SDK based on the following code. After the VBS backup is created, it will be displayed in the VBS list on the VBS console.

```
def create_backup(self):
    backup = {
        "volume_id": self.volume.id,
        "name": "sds",
        "description": "created by openstacksdk"
    }

    result = self.conn.volume_backup.create_backup(**backup)
    # assert result.job_id != None
    self.job_id = result.id
```


Table 3-4 Request parameter description

Parameter	Mandatory	Type	Description
backup	Yes	dict	Specifies the backup to be created.
volume_id	Yes	string	Specifies the ID of the disk to be backed up.
snapshot_id	No	string	Specifies the snapshot ID of the disk to be backed up.
name	Yes	string	Specifies the backup name. The value is a string of 1 to 64 characters consisting of digits, letters, underscores (_), and hyphens (-).
description	No	string	Provides supplementary information about the backup. The value is a string of 1 to 64 characters and cannot contain the less-than sign (<) or greater-than sign (>).

Querying VBS Backup Details

You can query the backup list and obtain the backup details using the Python OpenStack SDK based on the following code:

```
def query_backups_detail(self):
    backups = self.conn.volume_backup.backups(details=True)

    query = {
        "name": "volume-backup-" + self.volume.id,
        # "status": "available",
        "volume_id": self.volume.id,
        # "marker": "some-backup-id",
        "limit": 10
    }
    backups = self.conn.volume_backup.backups(details=True, **query)
    for backup in backups:
        print backup.name
```

Table 3-5 Request parameter description

Parameter	Mandatory	Type	Description
name	No	string	Specifies the name of the backup to be queried. This parameter is used to query the backups whose names are specified character strings.

Parameter	Mandatory	Type	Description
status	No	string	Specifies the status of the backup to be queried. This parameter is used to query the backups in a specified state. The value can be **available**, **error**, **restoring**, **creating**, **deleting**, or **error_deleting**.
offset	No	int	Specifies the offset of the queried details.
limit	No	int	Specifies the maximum number of query results that can be returned.
volume_id	No	string	Specifies the disk ID of the backup to be queried. This parameter is used to query the backups for specific disks.

Restoring a Disk Using a VBS Backup

You can restore a disk from a VBS backup using the Python OpenStack SDK based on the following code:

```
def restore_backup(self):
    self.query_backups()
    return self.conn.volume_backup.restore_backup(self.backup_id, self.volume.id)
```

Table 3-6 Request parameter description

Parameter	Mandatory	Type	Description
restore	Yes	dict	Specifies the operation of restoring the disk using a backup.
backup_id	Yes	string	Specifies the ID of the backup used to restore a disk.
volume_id	Yes	string	Specifies the ID of the disk to be restored.

Deleting a Backup

You can delete a backup using the Python OpenStack SDK based on the following code:

```
def delete_backup(self):
    self.query_backups()
    self.conn.volume_backup.delete_backup(self.backup_id)
```


Table 3-7 Request parameter description

Parameter	Mandatory	Type	Description
tenant_id	Yes	string	Specifies the ID of the tenant.
backup_id	Yes	string	Specifies the ID of the backup used to restore a disk.

Creating a Backup Policy

You can create a backup policy using the Python OpenStack SDK based on the following code:

```
def create_policy(self):
    data = {
        "remain_first_backup_of_curMonth": True,
        "rentention_num": 10,
        "frequency": 1,
        "start_time": "12:00",
        "status": "ON"
    }
    volume_backup_name = "SDK-backup-test-1"
    policy = self.conn.volume_backup.create_backup_policy(volume_backup_name, **data)
    print policy
```

Table 3-8 Request parameter description

Parameter	Mandatory	Type	Description
backup_policy_name	Yes	string	Specifies the backup policy name. The name is a string of 1 to 64 characters consisting of letters, digits, underscores (_), and hyphens (-). It cannot start with **default**.
scheduled_policy	Yes	dict	Specifies details about the scheduling policy.
start_time	Yes	string	Specifies the backup start time, which needs to be converted into the local UTC time (on the hour only). The value is in **HH:mm** format.
frequency	No	integer	Specifies the backup interval (1 to 14 days). Select either this parameter or **week_frequency**. If you select both, this parameter is used.
week_frequency	No	list<dict>	Specifies on which days of each week backup jobs are executed. The value can be one or more of the following: SUN, MON, TUE, WED, THU, FRI, SAT

Parameter	Mandatory	Type	Description
rentention_num	No	integer	Specifies the retained number (minimum: 2) of backups. Select either this parameter or **rentention_day**. If you select both, this parameter is used.
rentention_day	No	integer	Specifies how many days backups are retained.
remain_first_backup_of_curMonth	Yes	string	Specifies whether to retain the first backup in the current month. The value can be **Y** or **N**.
status	Yes	string	Specifies the backup policy status. The value can be **ON** or **OFF**.

Deleting a Backup Policy

You can delete a backup policy using the Python OpenStack SDK based on the following code:

```
def delete_policy(self):
    policy_id = self.query_policies().id
    self.conn.volume_backup.delete_backup_policy(policy_id)
```

Table 3-9 Request parameter description

Parameter	Mandatory	Type	Description
tenant_id	Yes	string	Specifies the ID of the tenant.
policy_id	Yes	string	Specifies the ID of the policy.

Querying Backup Policies

You can query backup policies using the Python OpenStack SDK based on the following code:

```
def query_policies(self):
    policies = list(self.conn.volume_backup.backup_policies())
    if policies and len(policies) > 0:
        return policies[0]
```

3.13 KMS Python SDK Demo

Key Management Service (KMS) is a secure, reliable, and easy-to-use service that helps users centrally manage and safeguard their Customer Master Keys (CMKs).

KMS uses hardware security modules (HSMs) to protect CMKs. HSMs help you create and control CMKs with ease. All CMKs are protected by root keys in HSMs

to avoid leakage. KMS implements access control and log-based tracking on all operations on CMKs. With records of use of all CMKs, it meets your audit and regulatory compliance requirements.

Creating a CMK

You can create a CMK using the Python OpenStack SDK based on the following code:

```
def create_key(conn):
    key_dict = {
        "key_alias": "test-key-123-456789223", "realm": "123"
    }
    key = conn.kms.create_key(**key_dict)
```

Enabling a CMK

You can enable a disabled CMK using the Python OpenStack SDK based on the following code:

```
def enable_key(conn, key):
    # a string of key id or an object of Key
    print(conn.kms.enable_key(key))
```

Creating a DEK

You can create a DEK using the Python OpenStack SDK based on the following code:

```
def create_data_key(conn, key):
    data_key_dict={
    "datakey_length":"512"
    }
    print(conn.kms.create_datakey(key, **data_key_dict))
```

Encrypting a DEK

You can encrypt a DEK using the Python OpenStack SDK based on the following code (The **plain_text** value is obtained during DEK creation):

```
def encrypt_datakey(conn, key):
    params = {
        "plain_text":
"4c5062132d3b1b450d1aff4cd49bb828c09e602e3678b3c8d9be5429fa22be17439a1c7bd167e76d1be8f0cadd
a76940c98e4483bc32312534ce98db824329eb
",
        "datakey_plain_length": "64"
    }
    datakey = conn.kms.encrypt_datakey(key, **params)
    print(datakey)
```

3.14 Anti-DDoS Python SDK Demo

Querying Optional Anti-DDoS Defense Policies

You can query optional Anti-DDoS defense policies. Based on your service, you can select a policy for Anti-DDoS traffic cleaning.

Code example:


```
def list_config(conn):
    print("list anti-ddos confit")
print(conn.anti_ddos.query_config_list())
```

Enabling Anti-DDoS

You can enable Anti-DDoS traffic cleaning defense. Successfully invoking this API only means that the service node has received the enabling request. You need to use the task querying API to check the task execution status.

Code example:

```
def create_eip(conn):
    fip_dict = {'enable_L7': True,
                'traffic_pos_id': 1,
                'http_request_pos_id': 1,
                'cleaning_access_pos_id': 1,
                'app_type_id': 0}

    fip = conn.anti_ddos.create_floating_ip(FLOATING_IP_ID, **fip_dict)
print(fip)
```

Disabling Anti-DDoS

You can disable Anti-DDoS traffic cleaning defense. Successfully invoking this API only means that the service node has received the disabling request. You need to use the task querying API to check the task execution status.

Code example:

```
def delete_eip(conn):
    fip = conn.anti_ddos.get_floating_ip(FLOATING_IP_ID)
    conn.anti_ddos.delete_floating_ip(fip)
```

Querying Configured Anti-DDoS Defense Policies

You can query the configured Anti-DDoS defense policy of a specified EIP.

Code example:

```
def get_eip(conn):
    fip = conn.anti_ddos.get_floating_ip(FLOATING_IP_ID)
    print(fip)
```

Updating Anti-DDoS Defense Policies

You can update the Anti-DDoS defense policy of a specified EIP. Successfully invoking this API only means that the service node has received the update request. You need to use the task querying API to check the task execution status.

Code example:

```
def update_eip(conn):
    fip = conn.anti_ddos.get_floating_ip(FLOATING_IP_ID)

    fip_update_dict = {'enable_L7': False,
                       'traffic_pos_id': 1,
                       'http_request_pos_id': 1,
                       'cleaning_access_pos_id': 1,
                       'app_type_id': 0}
    ufip = conn.anti_ddos.update_floating_ip(fip, **fip_update_dict)
    print(ufip)
```


Querying Anti-DDoS Tasks

You can query the execution status of a specified Anti-DDoS configuration task.

Code example:

```
def query_task_status(conn):
    print(conn.anti_ddos.query_task_status(
        '228186d4-4aec-4c37-bae8-cb025aaf5770'))
```

Querying the List of EIP Defense Statuses

You can query the defense statuses of all EIPs, regardless whether an EIP has been bound to an ECS or not.

Code example:

```
def list_eips(conn):
    print("list eips by status")
    for l in conn.anti_ddos.floating_ips():
    print(l)
```

Querying the Defense Status of a Specified EIP

You can query the defense status of a specified EIP.

Code example:

```
def get_eip_status(conn):
    print(conn.anti_ddos.get_eip_status(FLOATING_IP_ID))
```

Querying the Traffic of a Specified EIP

You can query the traffic of a specified EIP in the last 24 hours. Traffic is detected at five-minute intervals.

Code example:

```
def get_eip_daily(conn):
    for d in conn.anti_ddos.list_eip_daily(FLOATING_IP_ID):
        print(d)
```

Querying Events of a Specified EIP

You can query events of a specified EIP in the last 24 hours. Events include cleaning and blackhole events, and the query delay is within five minutes.

Code example:

```
def get_eip_log(conn):
    for l in conn.anti_ddos.list_eip_log(FLOATING_IP_ID):
        print(l)
```

Querying Weekly Defense Statistics

You can query weekly defense statistics about all your EIPs, including the number of intercepted DDoS attacks, number of attacks, and ranking by the number of attacks.

Code example:


```
def get_eip_weekly(conn):
    print(conn.anti_ddos.get_eip_weekly('1006510306'))
```

Querying Alarm Configuration

You can query alarm configuration, such as whether a certain type of alarms will be received, and whether alarms are received through SMS messages or emails.

Code example:

```
def get_alert_config(conn):
    print(conn.anti_ddos.get_alert_config())
```

3.15 DMS Python SDK Demo

Distributed Message Service (DMS) is a message middleware service based on distributed, high-availability clustering technology. It provides reliable, scalable, fully managed queues for storing messages. DMS enables cloud applications to decouple from each other, achieving high cost-effectiveness.

Creating a Queue

You can create a queue using the Python OpenStack SDK based on the following code. After the queue is created, messages will be sent to this queue.

```
queue_dict = {
                'name': "dmsTestQueue" + self.timeStamp,
                'description': "dmsTestQueue" + self.timeStamp
}
q = conn.dms.create_queue(**queue_dict)
```

Creating a Consumer Group

You can create a consumer group using the Python OpenStack SDK based on the following code. After the consumer group is created, it can consume messages in the queue.

```
groupDict = {
           "groups": [
              {
                  "name": "dmsConsumeGroup" + self.timeStamp
              }
           ]
 }
group = conn.dms.create_groups(queue, **groupDict)
```

Producing Messages

You can produce messages using the Python OpenStack SDK based on the following code:

```
msgDict = {
           "messages": [
              {
                  "body": "testMsg" + self.timeStamp,
                  "attributes":
                  {
                      "attribute1": "value1",
                      "attribute2": "value2"
                  }
```



```
            }
        ]
}
conn.dms.send_messages(queue, **msgDict))
```

Consume Messages

You can consume messages using the Python OpenStack SDK based on the following code:

```
msgList =  conn.dms.consume_message(queue, group[0].id)
```

3.16 MRS Python SDK Demo

MapReduce Service (MRS) provides users with a complete controllable, enterprise-class, big data cloud service, ensuring the smooth running of big data components, such as Hadoop, Spark, HBase, Kafka, and Storm.

Creating a Cluster and Submitting the Job

You can create a cluster and submit the job using the Python OpenStack SDK based on the following code. After the cluster is created, it will be displayed on the cluster page of the MRS console.

```
vpc_id = vpc_id
vpc_name = vpc_name
subnet_id = subnet_id
subnet_name = subnet_name
keypair_name = keypair_name

cluster = {
"cluster_name": cluster_name,
"billing_type": billing_type,
"data_center": data_center,
"master_node_num": master_node_num,
"master_node_size": master_node_size,
"core_node_num": core_node_num,
"core_node_size": core_node_size,
"available_zone_id": available_zone_id,
"vpc": vpc,
"vpc_id": vpc_id,
"subnet_id": subnet_id,
"subnet_name": subnet_name,
"cluster_version": cluster_version,
"cluster_type": cluster_type,
"volume_type": volume_type,
"volume_size": volume_size,
"keypair": keypair,
"safe_mode": safe_mode,
"component_list": [{
"component_id": component_id,
"component_name": component_name
}]
}

job = {
"job_type": job_type,
"job_name": job_name,
"jar_path": jar_path,
"arguments": arguments,
"input": input,
"output": output,
"job_log": job_log,
"shutdown_cluster": shutdown_cluster,
```



```
"file_action": file_action,
"submit_job_once_cluster_run": submit_job_once_cluster_run,
"hql": hql,
"hive_script_path": hive_script_path
}
cluster = conn.map_reduce.create_cluster_and_run_job(cluster, job)
```

Adding Clustered Nodes

You can add nodes to a cluster using the Python OpenStack SDK based on the following code by specifying the cluster ID:

```
expand_node_amount = instances
conn.map_reduce.expand_cluster(id, expand_node_amount)
```

Querying Cluster Details

You can query details of a cluster using the Python OpenStack SDK based on the following code by specifying the cluster ID:

```
cluster = conn.map_reduce.get_cluster(id)
```

Terminating a Cluster

You can terminate a cluster using the Python OpenStack SDK based on the following code by specifying the cluster ID:

```
conn.map_reduce.delete_cluster(id)
```

Adding and Executing a Job

You can add a job and execute using the Python OpenStack SDK based on the following code. After the job is created, it will be displayed on the job page of the MRS console.

```
exe = {
"job_type": job_type,
"job_name": job_name,
"cluster_id": cluster_id,
"jar_path": jar_path,
"arguments": arguments,
"input": input,
"output": output,
"job_log": job_log,
"file_action": file_action,
"hql": hql,
"hive_script_path": hive_script_path
}
job_exe = conn.map_reduce.exe_job (**exe)
```

Querying the Job Exe Object List

You can query job exe object list using the Python OpenStack SDK based on the following code:

```
query = {
"cluster_id": cluster_id,
"job_name": job_name,
"page_size": page_size,
"current_page": current_page,
"state": state
}
executions = list(conn.map_reduce.job_exes(**query))
```


Querying Details of a Job Exe Object

You can query details of a job exe object using the Python OpenStack SDK based on the following code by specifying the object ID:

```
conn.map_reduce.get_job_exe(id)
```

Deleting a Job Execution Object

You can delete a job execution object using the Python OpenStack SDK based on the following code by specifying the object ID:

```
conn.map_reduce.delete_job_execution(id)
```

3.17 RDS Python SDK Demo

Relational Database Service (RDS) is a cloud-based web service that is reliable, scalable, easy to manage, and immediately ready for use.

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **RDS**.

For details about the parameters supported by SDK, see the **Relational Database Service API Reference**.

The RDS v3 Python SDK supports operations on the following resource objects:

Resource Object	Example Code
backup	**Code**
config	**Code**
instance	**Code**
log	**Code**

3.18 CDN Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of CDN APIs.

The CDN Python SDK supports operations on the following resource objects.

Resource Object	Example Code
Acceleration domain name	**Code**
Statistic	**Code**
Log	**Code**

Resource Object	Example Code
Preheating task	Code
Refreshing task	Code
Query task	Code

3.19 FGS Python SDK Demo

For details about the mapping between SDK and APIs, see section **FGS**.

For details about the parameters supported by SDK, see the **description** of FGS APIs.

The FGS V2 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
function	Code
trigger	Code

3.20 TMS Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by TMS SDK, see the **description**.

The TMS v1 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
Tag management	Code

3.21 EPS Python SDK Demo

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by EPS SDK, see the **description**.

The EPS v1 Python SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
EnterpriseProject management	Code

4 Go

4.1 Getting Started with Go SDK

Welcome to use HUAWEI CLOUD developer tools (Go SDK). Go SDK allows you to easily access cloud services using codes.

This tutorial describes how to install and use Go SDK and provides examples to help you quickly get started.

The Go SDK is developed based on Gophercloud.

Supported Cloud Services

The Go SDK supports the following cloud services:

- **Elastic Cloud Server (ECS)**
- **Elastic Volume Service (EVS)**
- **Virtual Private Cloud (VPC)**
- **Identity and Access Management (IAM)**
- **Image Management Service (IMS)**
- **Elastic Load Balancing (ELB)**
- **Auto Scaling (AS)**

Prerequisites

1. You have obtained a cloud platform account and provisioned all required services.
2. If Go SDK is used, Go 1.9.1 is recommended.

SDK Acquisition and Installation

You can download the source code of Go SDK at the following Github websites:

https://github.com/huaweicloud/huaweicloud-sdk-release/tree/master/go-sdk

Perform the installation on the Linux OS:

Before the installation, ensure that the GOPATH environment variable points to the target directory of Gophercloud to be installed.

```
mkdir $HOME/go
mkdir -p $HOME/go/src
export GOPATH=$HOME/go
```

Download the source code package. Decompress it and install it in the **src** directory of the **go**. Run the **go build** command.

```
# unzip source code
unzip -d $GOPATH/src xxxxxx.zip
#run go build
cd $GOPATH/src/github.com/gophercloud/gophercloud
go build
```

How to Use

Before making a call to the SDK to access service APIs, you need to configure parameters and complete authentication. For details about the parameters, see **Table 4-1**.

```
package main

import (
    "github.com/gophercloud/gophercloud/auth/token"
    "github.com/gophercloud/gophercloud"
    "github.com/gophercloud/gophercloud/openstack/compute/v2/servers"
    "github.com/gophercloud/gophercloud/openstack"
    "fmt"
)

func main() {
// Set the authentication parameters.
    tokenOpts := token.TokenOptions{
        IdentityEndpoint: "https://iam.example.com/v3",
        Username:         "{username}",
        Password:         "{password}",
        DomainID:         "{domainid}",
        ProjectID:        "{projectid}",
    }
// Initialize the provider client.
    provider, providerErr := openstack.AuthenticatedClient(tokenOpts)
    if providerErr != nil {
        fmt.Println("init provider client error:", providerErr)
        panic(providerErr)
    }

// Initialize the service client.
    sc, serviceErr := openstack.NewComputeV2(provider, gophercloud.EndpointOpts{})
    if serviceErr != nil {
        fmt.Println("init compute service client error:", serviceErr)
        panic(serviceErr)
    }

// List all servers.
    allPages, err := servers.List(sc, servers.ListOpts{}).AllPages()

    if err != nil {
        fmt.Println("request server list error:", err)
        panic(err)
    }
// Parse the return values.
    allServers, err := servers.ExtractServers(allPages)
    if err != nil {
        fmt.Println("extract response data error:", err)
        if ue, ok := err.(*gophercloud.UnifiedError); ok {
```



```
            fmt.Println("ErrCode:", ue.ErrorCode())
            fmt.Println("Message:", ue.Message())
        }
        return
    }
// Print the information.
    fmt.Println("List Servers:")
    for _, s := range allServers {
        fmt.Println("server ID is :", s.ID)
        fmt.Println("server name is :", s.Name)
        fmt.Println("server Status is :", s.Status)
        fmt.Println("server AvailbiltyZone is :", s.AvailbiltyZone)
    }
}
```

- ProviderClient is the top-level client required by all OpenStack services. The client contains all authentication details, such as the URL and token ID. After the authentication, the compiled Go code can access the APIs.
- The Service Client of a service is required if you need to access this service. For details, see the related chapter about this service in this document.

Table 4-1 Parameter description

Parameter	Description	Example Value
IdentityEndpoint	Specifies the endpoint of the IAM service. **example** in https://iam.*example*.com/v3 indicates **Region.Cloud platform domain name**. For details about the parameter, see **here**.	https://iam.cn-north-1.myhuaweicloud.com/v3
Username	Specifies the IAM username. For details about how to obtain the username, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
Password	Specifies the IAM user password.	N/A
ProjectID	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
DomainID	Specifies the account ID. For details about how to obtain the account ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A

4.2 Using the Go SDK

4.2.1 Go SDK Authentication Modes

Go SDK supports two authentication modes: token-based authentication and AK/SK authentication.

Token Authentication

For details about the code for token-based authentication, see **Table 4-2**.

```
package main

import (
    "github.com/gophercloud/gophercloud/auth/token"
    "github.com/gophercloud/gophercloud"
    "github.com/gophercloud/gophercloud/openstack/compute/v2/servers"
    "github.com/gophercloud/gophercloud/openstack"
    "fmt"
)

func main() {
// Set the authentication parameters.
    tokenOpts := token.TokenOptions{
        IdentityEndpoint: "https://iam.example.com/v3",
        Username:         "{username}",
        Password:         "{password}",
        DomainID:         "{domainid}",
        ProjectID:        "{projectid}",
    }
// Initialize the provider client.
    provider, providerErr := openstack.AuthenticatedClient(tokenOpts)
    if providerErr != nil {
        fmt.Println("init provider client error:", providerErr)
        panic(providerErr)
    }

// Initialize the service client.
    sc, serviceErr := openstack.NewComputeV2(provider, gophercloud.EndpointOpts{})
    if serviceErr != nil {
        fmt.Println("init compute service client error:", serviceErr)
        panic(serviceErr)
    }

// List all servers.
    allPages, err := servers.List(sc, servers.ListOpts{}).AllPages()

    if err != nil {
        fmt.Println("request server list error:", err)
        panic(err)
    }
// Parse the return values.
    allServers, err := servers.ExtractServers(allPages)
    if err != nil {
        fmt.Println("extract response data error:", err)
        if ue, ok := err.(*gophercloud.UnifiedError); ok {
            fmt.Println("ErrCode:", ue.ErrorCode())
            fmt.Println("Message:", ue.Message())
        }
        return
    }
// Print the information.
    fmt.Println("List Servers:")
```



```
    for _, s := range allServers {
        fmt.Println("server ID is :", s.ID)
        fmt.Println("server name is :", s.Name)
        fmt.Println("server Status is :", s.Status)
        fmt.Println("server AvailbiltyZone is :", s.AvailbiltyZone)
    }
}
```

- ProviderClient is the top-level client required by all OpenStack services. The client contains all authentication details, such as the URL and token ID. After the authentication, the compiled Go code can access the APIs.
- The Service Client of a service is required if you need to access this service. For details, see the related chapter about this service in this document.

Table 4-2 Parameter description

Parameter	Description	Example Value
IdentityEndpoint	Specifies the endpoint of the IAM service. **example** in https://iam.*example*.com/v3 indicates **Region.Cloud platform domain name**. For details about the parameter, see **here**.	https://iam.cn-north-1.myhuaweicloud.com/v3
Username	Specifies the IAM username. For details about how to obtain the username, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
Password	Specifies the IAM user password.	N/A
ProjectID	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
DomainID	Specifies the account ID. For details about how to obtain the account ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A

AK/SK Authentication

For details about the code for AK/SK authentication, see **Table 4-3**.


```
package main

import (
    "github.com/gophercloud/gophercloud/auth/aksk"
    "github.com/gophercloud/gophercloud"
    "github.com/gophercloud/gophercloud/openstack/compute/v2/servers"
    "github.com/gophercloud/gophercloud/openstack"
    "fmt"
)

func main() {
// Set the authentication parameters.
    akskOpts := aksk.AKSKOptions{
        IdentityEndpoint: "https://iam.example.com/v3",
        DomainID:         "{domainid}",
        ProjectID:        "{projectid}",
        Cloud:            "myhuaweicloud.com",
        Region:           "cn-north-1",
        AccessKey:        "{your AK string}",
        SecretKey:        "{your SK string}",
    }
// Initialize the provider client.
    provider, providerErr := openstack.AuthenticatedClient(akskOpts)
    if providerErr != nil {
        fmt.Println("init provider client error:", providerErr)
        panic(providerErr)
    }

// Initialize the service client.
    sc, serviceErr := openstack.NewComputeV2(provider, gophercloud.EndpointOpts{})
    if serviceErr != nil {
        fmt.Println("init compute service client error:", serviceErr)
        panic(serviceErr)
    }

// List all servers.
    allPages, err := servers.List(sc, servers.ListOpts{}).AllPages()

    if err != nil {
        fmt.Println("request server list error:", err)
        panic(err)
    }
// Parse the return values.
    allServers, err := servers.ExtractServers(allPages)
    if err != nil {
        fmt.Println("extract response data error:", err)
        if ue, ok := err.(*gophercloud.UnifiedError); ok {
            fmt.Println("ErrCode:", ue.ErrorCode())
            fmt.Println("Message:", ue.Message())
        }
        return
    }
// Print the information.
    fmt.Println("List Servers:")
    for _, s := range allServers {
        fmt.Println("server ID is :", s.ID)
        fmt.Println("server name is :", s.Name)
        fmt.Println("server Status is :", s.Status)
        fmt.Println("server AvailbiltyZone is :", s.AvailbiltyZone)
    }
}
```

AK/SK generation description: Log in to the management console, choose **My Credentials**, and click **Access Keys** to create an AK and SK.

The time error between the AK/SK signature time and UTC time cannot exceed 15 minutes. Otherwise, the authentication fails.

If the AK/SK signature fails for more than five consecutive times, the AK/SK request of the source IP address is locked for 5 minutes.

Table 4-3 Parameter description

Parameter	Description	Example Value
IdentityEndpoint	Specifies the endpoint of the IAM service. **example** in https://iam.*example*.com/v3 indicates **Region.Cloud platform domain name**. For details about the parameter, see **here**.	https://iam.cn-north-1.myhuaweicloud.com/v3
DomainID	Specifies the account ID. For details about how to obtain the account ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
ProjectID	Specifies the project ID. For details about how to obtain the project ID, see **How Do I Obtain the IAM Username, Account ID, and Project ID?**.	N/A
ak/sk	Specifies the AK/SK access key. **NOTE** • AK/SK generation description: Log in to the management console, choose **My Credentials**, and click **Access Keys** to create an AK and SK. • The time error between the AK/SK signature time and UTC time cannot exceed 15 minutes. Otherwise, the authentication fails. • If the AK/SK signature fails for more than five consecutive times, the AK/SK request of the source IP address is locked for 5 minutes.	N/A
Region	Specifies the region name.	cn-north-1

Parameter	Description	Example Value
Cloud	Specifies the cloud platform domain name.	myhuaweicloud.com

4.2.2 Go SDK Troubleshooting

To enable the debugging function using Go SDK, add the following code to the application:

```
import (
	"github.com/gophercloud/gophercloud"
)
gophercloud.EnableDebug=true
```

4.3 ECS Go SDK User Guide

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of ECS APIs.

The ECS v1 go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
cloudservers	**Code**
tags	**Code**
flavors	**Code**

The ECS v1.1 go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
cloudservers	**Code**

The ECS v2 go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code

cloudservers	**Code**

The Nova v2 go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
flavors	**Code**
images	**Code**
keypairs	**Code**
quotas	**Code**
servers	**Code**
volume attach	**Code**

4.4 EVS Go SDK User Guide

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of EVS APIs.

The Cinder v2 Go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
volume	**Code**

4.5 VPC Go SDK User Guide

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by the SDK when the VPC SDK is used, see the **description** of VPC APIs.

The VPC v1 Go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
bandwidth	Code
port	Code
private ip	Code
public ip	Code
quota	Code
security group	Code
security group rule	Code
subnet	Code
vpc	Code

4.6 IAM Go SDK User Guide

Service Client

To interact with OpenStack APIs, transfer the identity credentials to the provider. After that, if you want to retrieve any information about the identity service, you need to make a call to NewIdentityV3 to create an identity service client and then use required SDKs.

```
client, err := openstack.NewIdentityV3(provider, gophercloud.EndpointOpts{
 Region: "RegionName",
})
```

Tokens

A token is a text returned during authentication. It is used for subsequent access and control of API resources. Each token has a specific range, which specifies the resources that can be accessed using the token.

Create a token.

```
import (    "github.com/gophercloud/gophercloud/openstack/identity/v3/tokens"    )

func TestCreateTokenByPasswd(t *testing.T){
    scope := tokens.Scope{
        ProjectName:  TenantName,
        DomainID:     DomainID,
    }

    authOptions := tokens.AuthOptions{
        Username:   Username,
        Password:   Password,
        DomainName: DomainName,
        Scope:      scope,
    }


    token, err := tokens.Create(client, &authOptions).Extract()
}
```


Validate a token.

Check whether the token is valid.

```
func TestValidateToken(t *testing.T) {

    token, err := tokens.Validate(client, "token_id")
}
```

Service Catalogs

A service is a RESTful API that controls an OpenStack service function. The controlled OpenStack services include the compute and object storage services. A service provides one or multiple endpoints. Through these endpoints, you can access resources and perform required operations.

List services.

```
import (    "github.com/gophercloud/gophercloud/openstack/identity/v3/services"    )

func TestServicesList(t *testing.T) {
// List enumerates the services available to a specific user.
listOpts := services.ListOpts{
        ServiceType: "identity",
    }
}
// Retrieve a pager (i.e. a paginated collection)
allPages, err := services.List(client, listOpts).AllPages()

// Define an anonymous function to be executed on each page's iteration
allServices, err := services.ExtractServices(allPages)

for _, service := range allServices {
    // " service " will be a services.Service
}
```

Get details about a specific service.

```
func TestServicesGet(t *testing.T) {
    service, err := services.Get(client,"service_id").Extract()
}
```

Endpoints

An endpoint is a network accessible address. Generally, it is a URL address used to access a service.

List endpoints.

```
import (    "github.com/gophercloud/gophercloud/openstack/identity/v3/endpoints"    )

func TestEndpointsList(t *testing.T) {
    endpointListOpts := endpoints.ListOpts{
        Availability: gophercloud.AvailabilityPublic,
    }
// Retrieve a pager (i.e. a paginated collection)
    allPages, err := endpoints.List(client, endpointListOpts).AllPages()

    // Define an anonymous function to be executed on each page's iteration
allEndpoints, err := endpoints.ExtractEndpoints(allPages)

    for _, endpoint := range allEndpoints {
        // "endpoint " will be a endpoints.Endpoint
    }
}
```


4.7 IMS Go SDK User Guide

Service Client

To interact with OpenStack APIs, transfer the identity credentials to the provider. After that, if you want to retrieve any information about the image service, you need to make a call to NewImageServiceV2 to create an image service client and then use required SDKs.

```
client, err := openstack.NewImageServiceV2(provider, gophercloud.EndpointOpts{
 Region: "RegionName",
})
```

Images

An image is the OS of a VM and it is a series of files used to create or rebuild servers. By default, the carrier provides preset OS images, but you can also create customized images from the ECSs.

Create an image.

You can use Create to create an image.

```
import (    "github.com/gophercloud/gophercloud/openstack/imageservice/v2/images"    )

func TestImagesCreateDestroyEmptyImage(t *testing.T) {
    protected := false
    visibility := images.ImageVisibilityPrivate
    createOpts := &images.CreateOpts{
          Name:             name,
          ContainerFormat:   "bare",
          DiskFormat:       "vhd",
          MinDisk:          40,
          MinRAM:           1024,
          Protected:        &protected,
          Visibility:       &visibility,
          Tags:             []string{"test","adsfi"},
          Properties: map[string]string{
                "architecture": "x86_64",
          },
    }
    image, err := images.Create(client, createOpts).Extract()
}
```

Update an image.

```
func TestImagesUpdate(t *testing.T) {
    updateOpts := images.UpdateOpts{
          images.ReplaceImageName{
              NewName: "alternateName",
          },
    }
    image, err := images.Update(client, "imageupdate_id", updateOpts).Extract()
}
```

List images.

```
func TestImagesListALL(t *testing.T) {
    listOpts := images.ListOpts{
        Visibility: images.ImageVisibilityPublic,
        Owner:      "owner_id",
        Status:     "active",
        Marker:     "marker_id",
```



```
        SortKey:    "name",
        SortDir:    "asc",
    }

    // Retrieve a pager (i.e. a paginated collection)
    allPages, err := images.List(client, listOpts).AllPages()

    // Define an anonymous function to be executed on each page's iteration
    allImages, err := images.ExtractImages(allPages)

    for _, image := range allImages {
    // " image " will be a images.Image
    }
}
```

Get details about a specific image.

```
func TestImagesGet(t *testing.T) {
    image, err := images.Get(client, "image_id").Extract()
}
```

Delete an image.

```
func DeleteImage(t *testing.T, client *gophercloud.ServiceClient, image *images.Image) {
    err := images.Delete(client, "image_id").ExtractErr()
}
```

4.8 ELB Go SDK User Guide

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **A.2.9 ELB**.

For details about the parameters supported by SDK, see the **description** of ELB APIs.

The Enhanced Load Balancer (ELB) v2 Go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
loadbalacner	**Code**
listener	**Code**
pool	**Code**
member	**Code**
healthmonitor	**Code**
certificate	**Code**
l7policy	**Code**
l7rule	**Code**
whitelist	**Code**

4.9 AS Go SDK User Guide

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **AS**.

For details about the parameters supported by SDK, see the **description** of AS APIs.

The AS v1 Go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
configration	**Code**
group	**Code**
instance	**Code**
lifecyclehook	**Code**
log	**Code**
notification	**Code**
policy	**Code**
policylog	**Code**
quota	**Code**
tag	**Code**

4.10 Cloud Eye Go SDK User Guide

The SDK corresponds to RESTful APIs. For details about the APIs supported by the latest SDK version and the mapping between the SDK and APIs, see **here**.

For details about the parameters supported by SDK, see the **description** of Cloud Eye APIs.

The Cloud Eye V1.0 Go SDK supports the following resource. The table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
metric	**Code**
alarm	**Code**
metric data	**Code**
quota	**Code**

4.11 FGS Go SDK User Guide

For details about the mappings between SDKs and APIs, see **FGS**.

For details about the parameters supported by SDK, see the **description** of FGS APIs.

The FGS V2 Go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

Resource Object	Example Code
function	**Code**
trigger	**Code**

4.12 RDS Go SDK User Guide

For details about the mapping between SDKs and APIs, see **RDS**.

For details about the parameters supported by SDK, see the **Relational Database Service API Reference**.

The RDS v3 Go SDK supports operations on the following resource objects. The following table lists the example code for adding, modifying, deleting, and querying a resource.

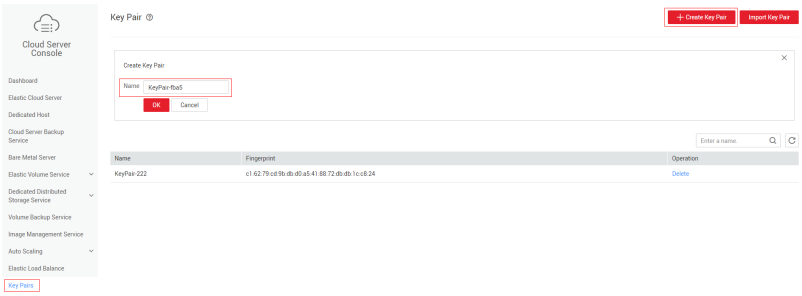
Resource Object	Example Code
datastore	**Code**
instance	**Code**
configuration	**Code**
backup	**Code**
database	**Code**
dbuser	**Code**
flavor	**Code**

5 SDK-related FAQ

5.1 How Can I Create a Key Pair on the Console?

Click **Create SSH Key Pair** to create a key pair **myOpenStackKey**, click **OK**, and save the **myOpenStackKey.pem** file to the local PC.

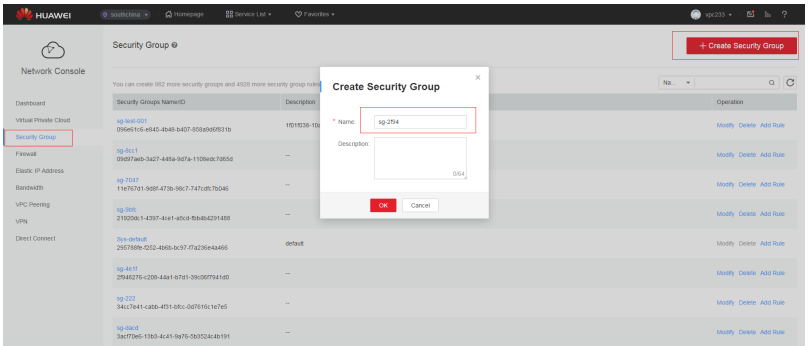
Figure 5-1 Creating a key pair



5.2 How Can I Create a Security Group?

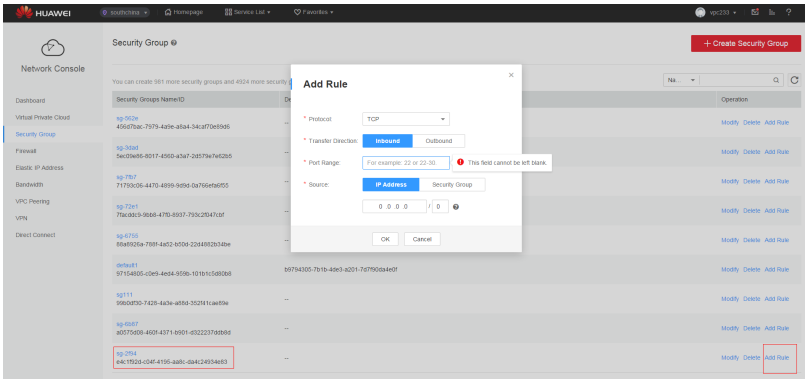
1. Choose **Security Group** > **Create Security Group**.

Figure 5-2 Creating a security group



2. Click **Add Rule**. On the displayed dialog box, add rules.

Figure 5-3 Adding a security group rule



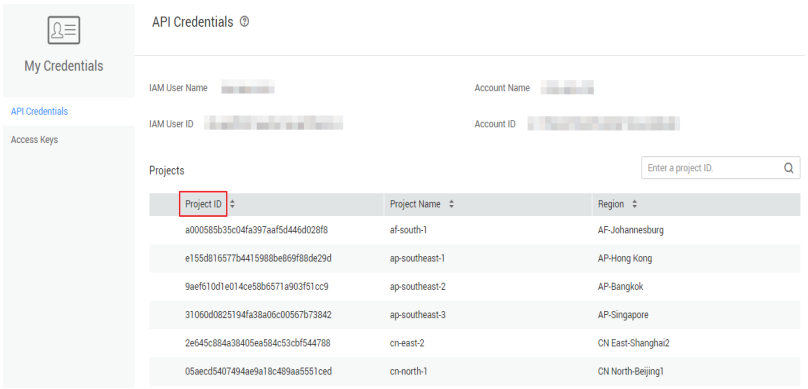
5.3 How Do I Obtain the IAM Username, Account ID, and Project ID?

This section describes how to obtain the IAM username, account ID, and project ID on the management console.

1. Log in to the management console.
2. Hover the mouse over the username in the upper right corner and select **My Credentials** from the drop-down list.

   On the displayed **API Credentials** page, view the IAM username, account ID, and project ID.

Figure 5-4 Viewing required information on the **API Credentials** page



5.4 Service Name List

Service Name Abbreviation	Service Name	Remarks
ECS	Elastic Cloud Server	-

Service Name Abbreviation	Service Name	Remarks
EVS	Elastic Volume Service	-
VPC	Virtual Private Cloud	-
IMS	Image Management Service	-
IAM	Identity and Access Management	-
CES	Cloud Eye Service	-
CTS	Cloud Trace Service	-
DNS	Domain Name Service	-
AS	Auto Scaling	-
ELB	Elastic Load Balance	-
VBS	Volume Backup Service	-

6 Mappings Between APIs and SDKs

6.1 Java

6.1.1 IAM

The SDK interfaces based on the IAM API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
TokenService	Token create(String nocatalog, Auth auth)	POST /v3/auth/tokens Obtaining a User Token Through Password Authentication **Link** Obtaining a User Token Through Password and Virtual MFA Authentication **Link** Obtaining an Agency Token **Link**
	Token get(String tokenId) Token getWithoutCatalog(String tokenId, String nocatalog	GET /v3/auth/tokens **Link**
	List<? extends Project> getProjectScopes(String tokenId)	GET /v3/auth/projects **Link**
	List<? extends Domain> getDomainScopes(String tokenId)	GET /v3/auth/domains **Link**
	List<? extends Service> getServiceCatalog()	GET /v3/auth/catalog **Link**

Interface	Method	API
SecuritytokenService	Securitytoken create(Auth auth)	POST /v3.0/OS-CREDENTIAL/securitytokens **Link**
CredentialService	PermanentCredentialResp createPermanentAccessKey(CreatePermanentCredentialReq createPermanentCredentialReq)	POST /v3.0/OS-CREDENTIAL/credentials **Link**
	Credentials listPermanentAccessKeys(String userId)	GET /v3.0/OS-CREDENTIAL/credentials **Link**
	PermanentCredentialResp queryPermanentAccessKey(String accessKey)	GET /v3.0/OS-CREDENTIAL/credentials/{access_key} **Link**
	UpdateCredentialResp updatePermanentAccessKey(String accessKey, UpdateCredentialReq updateCredentialReq)	PUT /v3.0/OS-CREDENTIAL/credentials/{access_key} **Link**
	ActionResponse deletePermanentAccessKey(String accessKey)	DELETE /v3.0/OS-CREDENTIAL/credentials/{access_key} **Link**
RegionService	List<? extends Region> list()	GET /v3/regions **Link**
	Region get(String regionId)	GET /v3/regions/{region_id} **Link**
ProjectService(identity)	List<? extends Project> listByObject(Map<String, Object> filteringParams)	GET /v3/projects **Link**
	Project create(Project project)	POST /v3/projects **Link**
	Project update(Project project)	PATCH /v3/projects/{project_id} **Link**
	Project get(String projectId)	GET /v3/projects/{project_id} **Link**
ProjectService(iam)	ActionResponse updateStatus(String projectId, UpdateProjectReq project)	PUT /v3-ext/projects/{project_id} **Link**

Interface	Method	API
	QueryProjectResp query(String projectId)	GET /v3-ext/projects/{project_id} **Link**
DomainService	PasswordConfig getDomainPasswordConfig(String domainId)	GET /v3/domains/{domain_id}/config/security_compliance **Link**
	PasswordConfig getDomainPasswordConfigByOption(String domainId, String option)	GET /v3/domains/{domain_id}/config/security_compliance/{option} **Link**
UserService(identity)	List<? extends User> list() List<? extends User> list(Map<String, String> filteringParams)	GET /v3/users **Link**
	User get(String userId)	GET /v3/users/{user_id} **Link**
	List<? extends Group> listUserGroups(String userId)	GET /v3/users/{user_id}/groups **Link**
	User create(User user)	POST /v3/users **Link**
	ActionResponse changePassword(String userId,String originalPassword,String password)	POST /v3/users/{user_id}/password **Link**
	User update(User user)	PATCH /v3/users/{user_id} **Link**
	ActionResponse delete(String userId)	DELETE /v3/users/{user_id} **Link**
	List<? extends Project> listUserProjects(String userId)	GET /v3/users/{user_id}/projects **Link**
UserService(iam)	CreateUserResp create(CreateUserReq user)	POST /v3.0/OS-USER/users **Link**
	QueryUserResp query(String userId)	GET /v3.0/OS-USER/users/{user_id} **Link**

Interface	Method	API
	ActionResponse update(String userId, UpdateUserReq user)	PUT /v3.0/OS-USER/users/{user_id}/info **Link**
	UpdateUserByAdminResp updateByAdmin(String userId, UpdateUserByAdminReq user)	PUT /v3.0/OS-USER/users/{user_id} **Link**
GroupService	List<? extends Group> list(Map<String, String> filteringParams)	GET /v3/groups **Link**
	Group get(String groupId)	GET /v3/groups/{group_id} **Link**
	Group create(Group group)	POST /v3/groups **Link**
	Group update(Group group)	PATCH /v3/groups/{group_id} **Link**
	ActionResponse delete(String groupId)	DELETE /v3/groups/{group_id} **Link**
	ActionResponse checkGroupUser(String groupId, String userId)	HEAD /v3/groups/{group_id}/users/{user_id} **Link**
	ActionResponse addUserToGroup(String groupId, String userId)	PUT /v3/groups/{group_id}/users/{user_id} **Link**
	ActionResponse removeUserFromGroup(String groupId, String userId)	DELETE /v3/groups/{group_id}/users/{user_id} **Link**
	List<? extends User> listGroupUsers(String groupId, Map<String, String> filteringParams)	GET /v3/groups/{group_id}/users **Link**
	List<? extends Role> listDomainGroupRoles(String groupId, String domainId)	GET /v3/domains/{domain_id}/groups/{group_id}/roles **Link**
	List<? extends Role> listProjectGroupRoles(String groupId, String projectId)	GET /v3/projects/{project_id}/groups/{group_id}/roles **Link**

Interface	Method	API
RoleService	List<? extends Role> list(Map<String, String> filteringParams)	GET /v3/roles **Link**
	Role get(String roleId)	GET /v3/roles/{role_id} **Link**
	ActionResponse grantDomainGroupRole(String domainId, String groupId, String roleId)	PUT /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id} **Link**
	ActionResponse grantProjectGroupRole(String projectId, String groupId, String roleId)	PUT /v3/projects/{project_id}/groups/{group_id}/roles/{role_id} **Link**
	ActionResponse checkDomainGroupRole(String domainId, String groupId, String roleId)	HEAD /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id} **Link**
	ActionResponse checkProjectGroupRole(String projectId, String groupId, String roleId)	HEAD /v3/projects/{project_id}/groups/{group_id}/roles/{role_id} **Link**
	ActionResponse revokeDomainGroupRole(String domainId, String groupId, String roleId)	DELETE /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id} **Link**
	ActionResponse revokeProjectGroupRole(String projectId, String groupId, String roleId)	DELETE /v3/projects/{project_id}/groups/{group_id}/roles/{role_id} **Link**
	ActionResponse grantGroupAllProjectsRole(String domainId, String groupId, String roleId)	PUT /v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects **Link**
CustomRoleService	List<CreateRoleResp> list().getRoles()	GET /v3.0/OS-ROLE/roles **Link**
	QueryRoleResp get(String roleId)	GET /v3.0/OS-ROLE/roles/{role_id} **Link**

Interface	Method	API
	CreateRoleResp create(CreateRoleReq roleReq)	POST /v3.0/OS-ROLE/roles **Link** for creating a custom cloud service policy **Link** for creating a custom agency policy
	UpdateRoleResp update(String roleId, UpdateRoleReq updateRoleReq)	PATCH /v3.0/OS-ROLE/roles/{role_id} **Link** for modifying a custom cloud service policy **Link** for modifying a custom agency policy
	ActionResponse delete(String roleId)	DELETE /v3.0/OS-ROLE/roles/{role_id} **Link**
AgencyService	ListAgenciesResp listAgencies(String domainId, Map<String, String> filteringParams)	GET /v3.0/OS-AGENCY/agencies **Link**
	AgencyResp get(String agencyId)	GET /v3.0/OS-AGENCY/agencies/{agency_id} **Link**
	AgencyResp create(CreateAgencyReq createAgencyReq)	POST /v3.0/OS-AGENCY/agencies **Link**
	AgencyResp update(String agencyId, UpdateAgencyReq updateAgencyReq)	PUT /v3.0/OS-AGENCY/agencies/{agency_id} **Link**
	ActionResponse delete(String agencyId)	DELETE /v3.0/OS-AGENCY/agencies/{agency_id} **Link**
	ListPermissionsResp listPermissionsOnDomain(String domainId, String agencyID)	GET /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles **Link**
	ListPermissionsResp listPermissionsOnProject(String projectId, String agencyId)	GET /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles **Link**

Interface	Method	API
	ActionResponse addPermissionOnDomain(String domainId, String agencyId, String roleId)	PUT /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id} **Link**
	ActionResponse addPermissionOnProject(String projectId, String agencyId, String roleId)	PUT /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id} **Link**
	ActionResponse checkPermissionOnDomain(String domainId, String agencyId, String roleId)	HEAD /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id} **Link**
	ActionResponse checkPermissionOnProject(String projectId, String agencyId, String roleId)	HEAD /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id} **Link**
	ActionResponse deletePermissionOnDomain(String domainId, String agencyId, String roleId)	DELETE /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id} **Link**
	ActionResponse deletePermissionOnProject(String projectId, String agencyId, String roleId)	DELETE /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id} **Link**
VersionService	Version get()	GET /v3 **Link**
ServiceEndpointService	List<? extends Service> list(Map<String, String> filteringParams)	GET /v3/services **Link**
	Service get(String serviceId)	GET /v3/services/{service_id} **Link**
	List<? extends Endpoint> listEndpoints(Map<String, String> filteringParams)	GET /v3/endpoints **Link**
	Endpoint getEndpoint(String endpointId)	GET /v3/endpoints/{endpoint_id} **Link**

6.1.2 IMS

The SDK interfaces based on the Glance v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
ImageService	Image create(Image image)	POST /v2/images
	ActionResponse upload(String imageId, Payload payload, Image image)	PUT /v2/images/{image_id}/file
	ActionResponse delete(String imageId)	DELETE /v2/images/{image_id}
	List<? extends Image> list()	GET /v2/images
	Image get(String imageId)	GET /v2/images/{image_id}
	ActionResponse updateTag(String imageId, String tagkeyvalue)	PUT /v2/images/{image_id}/tags/{tag}
	ActionResponse deleteTag(String imageId, String tagkey)	DELETE /v2/images/{image_id}/tags/{tag}
	List<? extends Member> listMembers(String imageId)	GET /v2/images/{image_id}/members
	Member getMember(String imageId, memberId)	GET /v2/images/{image_id}/members/{member_id}
	ActionResponse deleteMember(String imageId, String memberId)	DELETE /v2/images/{image_id}/members/{member_id}
	Member updateMember(String imageId, String memberId, Member.MemberStatus.ACCEPTED)	PUT /v2/images/{image_id}/members/{member_id}
	Member createMember(String imageId, String memberId)	POST /v2/images/{image_id}/members

The SDK interfaces based on the IMS v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
ImageService	String create(ImageCreateByInstance imageCreateByInstance)	POST /v2/cloudimages/action

Interface	Method	API
	String create(ImageCreateByExternalImage imageCreateByExternalImage)	POST /v2/cloudimages/action
	List<Image> list(Map<String, String> filteringParams)	GET /v2/cloudimages
	Image update(List<ImageUpdate> updateModel, String imageId)	PATCH /v2/cloudimages/{image_id}

The SDK interfaces based on the IMS v1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
ImageService	String create(ImageCreateByOBS imageCreateByOBS)	POST /v1/cloudimages/dataimages/action
	String regist(RegistImage image, String imageId)	PUT /v1/cloudimages/{image_id}/upload
	String export(ExportImage image, String imageId)	POST /v1/cloudimages/{image_id}/file
	String create(ImageCreateByExternalImage imageCreateByExternalImage)	POST /v1/cloudimages/wholeimages/action
	String create(ImageCreateByInstance imageCreateByInstance)	POST /v1/cloudimages/wholeimages/action
Jobservice	Job get(String jobId)	GET /v1/{project_id}/jobs/{job_id}

6.1.3 VPC

The SDK interfaces based on the VPC v1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
VpcService	Vpc create(VpcCreate creation)	POST /v1/{project_id}/vpcs **Link**
	Vpc get(String vpcId)	GET /v1/{project_id}/vpcs/{vpc_id} **Link**

Interface	Method	API
	List<? extends Vpc> list()	GET /v1/{project_id}/vpcs **Link**
	List<? extends Vpc> list(Map<String, String> filteringParams)	GET /v1/{project_id}/vpcs **Link**
	Vpc update(String vpcId, VpcUpdate vpcUpdate)	PUT /v1/{project_id}/vpcs/{vpc_id} **Link**
	ActionResponse delete(String vpcId)	DELETE /v1/{project_id}/vpcs/{vpc_id} **Link**
PublicIpService	VirtualPublicIpsResp apply(VirtualPublicIps virtualPublicIps)	POST /v1/{project_id}/publicips **Link**
	VirtualPublicIp get(String publicipId)	GET /v1/{project_id}/publicips/{publicip_id} **Link**
	List<? extends PublicIp> list()	GET /v1/{project_id}/publicips **Link**
	List<? extends PublicIp> list(Map<String, String> filteringParams)	GET /v1/{project_id}/publicips **Link**
	PublicIp update(String publicIpId, PublicIpUpdate publicIpUpdate)	PUT /v1/{project_id}/publicips/{publicip_id} **Link**
	ActionResponse delete(String publicIpId)	DELETE /v1/{project_id}/publicips/{publicip_id} **Link**
PrivateIpService	List<? extends PrivateIp> apply(PrivateIps privateIps)	POST /v1/{project_id}/privateips **Link**
	PrivateIp get(String privateIpId)	GET /v1/{project_id}/privateips/{privateip_id} **Link**

Interface	Method	API
	List<? extends PrivateIp> list(String subnetId)	GET /v1/{project_id}/subnets/{subnet_id}/privateips **Link**
	List<? extends PrivateIp> list(String subnetId, Map<String, String> filteringParams)	GET /v1/{project_id}/subnets/{subnet_id}/privateips **Link**
	ActionResponse delete(String privateIpId)	DELETE /v1/{project_id}/privateips/{privateip_id} **Link**
SecurityGroupService	SecurityGroup create(SecurityGroupCreate securityGroup)	POST /v1/{project_id}/security-groups **Link**
	SecurityGroup get(String securityGroupId)	GET /v1/{project_id}/security-groups/{security_group_id} **Link**
	List<? extends SecurityGroup> list()	GET /v1/{project_id}/security-groups **Link**
	List<? extends SecurityGroup> list(Map<String, String> filteringParams)	GET /v1/{project_id}/security-groups **Link**
	ActionResponse delete(String securityGroupId)	DELETE /v1/{project_id}/security-groups/{security_group_id} **Link**
	SecurityGroupRule createSecurityGroupRule(SecurityGroupRule securityGroupRule)	POST /v1/{project_id}/security-group-rules **Link**
	SecurityGroupRule getSecurityGroupRule(String securityGroupRuleId)	GET /v1/{project_id}/security-group-rules/{rules_security_groups_id} **Link**
	List<? extends SecurityGroupRule> listSecurityGroupRules()	GET /v1/{project_id}/security-group-rules **Link**

Interface	Method	API
	List<? extends SecurityGroupRule> listSecurityGrou-pRules(Map<String, String> filteringParams)	GET /v1/{project_id}/security-group-rules **Link**
	ActionResponse deleteSecurityGrou-pRule(String securityGroupRuleId)	DELETE /v1/{project_id}/security-group-rules/{rules_security_groups_id} **Link**
BandWidthService	VirtualBandWidths get(String bandwidthId)	GET /v1/{project_id}/bandwidths/{bandwidth_id} **Link**
	List<VirtualBandWidths> list()	GET /v1/{project_id}/bandwidths **Link**
	List<VirtualBandWidths> list(Map<String, String> filteringParams)	GET /v1/{project_id}/bandwidths **Link**
	VirtualBandWidths update(VirtualBandWidthUpdate bandWidtUpdate, String bandwidthId)	PUT /v1/{project_id}/bandwidths/{bandwidth_id} **Link**
QuotaService	Quotas list()	GET /v1/{project_id}/quotas **Link**
	Quotas list(String type)	GET /v1/{project_id}/quotas **Link**
SubnetService	Subnet create(SubnetCreate creation)	POST /v1/{project_id}/subnets **Link**
	Subnet get(String subnetId)	GET /v1/{project_id}/subnets/{subnet_id} **Link**
	List<? extends Subnet> list()	GET /v1/{project_id}/subnets **Link**

Interface	Method	API
	List<? extends Subnet> list(Map<String, String> filteringParams)	GET /v1/{project_id}/subnets **Link**
	SubnetUpdateResp update(String vpcId, String subnetId, SubnetUpdate subnetUpdate)	PUT /v1/{project_id}/vpcs/{vpc_id}/subnets/{subnet_id} **Link**
	ActionResponse delete(String vpcId,String subnetId)	DELETE /v1/{project_id}/vpcs/{vpc_id}/subnets/{subnet_id} **Link**
PortService	Port create(PortCreate port)	POST /v1/{project_id}/ports **Link**
	Port get(String portId)	GET /v1/{project_id}/ports/{port_id} **Link**
	List<? extends Port> list()	GET /v1/{project_id}/ports **Link**
	List<? extends Port> list(Map<String, String> filteringParams)	GET /v1/{project_id}/ports **Link**
	Port update(String portId, PortUpdate portUpdate)	PUT /v1/{project_id}/ports/{port_id} **Link**
	ActionResponse delete(String portId)	DELETE /v1/{project_id}/ports/{port_id} **Link**

The SDK interfaces based on the VPC v2.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
PublicIpService	AsyncPublicipRespEntity apply(VirtualPublicIps virtualPublicIps)	POST /v2.0/{project_id}/publicips **Link**

Interface	Method	API
BandWidthService	AsyncBandWidthRespEntity update(VirtualBandWidths bandWidth, String bandwidthId)	PUT /v2.0/{project_id}/bandwidths/{bandwidth_id} **Link**

The SDK interfaces based on the Neutron v2.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
NetFloatingIPService	NetFloatingIP associateToPort(String id, String portId)	PUT /v2.0/floatingips/{floatingip_id} **Link**
	NetFloatingIP create(NetFloatingIP floatingIp)	POST /v2.0/floatingips **Link**
	ActionResponse delete(String id)	DELETE /v2.0/floatingips/{floatingip_id} **Link**
	NetFloatingIP disassociateFromPort(String id)	PUT /v2.0/floatingips/{floatingip_id} **Link**
	NetFloatingIP get(String id)	GET /v2.0/floatingips/{floatingip_id} **Link**
	List<? extends NetFloatingIP> list()	GET /v2.0/floatingips **Link**
	List<? extends NetFloatingIP> list(Map<String, String> filteringParams)	GET /v2.0/floatingips **Link**
NetworkService	Network create(Network network)	POST /v2.0/networks **Link**
	ActionResponse delete(String networkId)	DELETE /v2.0/networks/{network_id} **Link**
	Network get(String networkId)	GET /v2.0/networks/{network_id} **Link**

Interface	Method	API
	List<? extends Network> list()	GET /v2.0/networks **Link**
	List<? extends Network> list(Map<String, String>filteringParams)	GET /v2.0/networks **Link**
	Network update(String networkId, NetworkUpdate network)	PUT /v2.0/networks/{network_id} **Link**
PortService	Port create(Port port)	POST /v2.0/ports **Link**
	ActionResponse delete(String portId)	DELETE /v2.0/ports/{port_id} **Link**
	Port get(String portId)	GET /v2.0/ports/{port_id} **Link**
	List<? extends Port> list()	GET /v2.0/ports **Link**
	List<? extends Port> list(PortListOptions options)	GET /v2.0/ports?network_id={network_id} **Link**
	Port update(Port port)	PUT /v2.0/ports/{port_id} **Link**
RouterService	RouterInterface attachInterface(String routerId, AttachInterface-Type type, String portOrSubnetId)	PUT /v2.0/routers/{router_id}/add_router_interface **Link**
	Router create(Router router)	POST /v2.0/routers **Link**
	Router create(String name, boolean adminStateUp)	POST /v2.0/routers **Link**
	ActionResponse delete(String routerId)	DELETE /v2.0/routers/{router_id} **Link**

Interface	Method	API
	RouterInterface detachInterface(String routerId, String subnetId, String portId)	PUT /v2.0/routers/{router_id}/remove_router_interface **Link**
	Router get(String routerId)	GET /v2.0/routers/{router_id} **Link**
	List<? extends Router>list()	GET /v2.0/routers **Link**
	Router toggleAdminStateUp(String routerId, boolean adminStateUp)	PUT /v2.0/routers/{router_id} **Link**
	Router update(Router router)	PUT /v2.0/routers/{router_id} **Link**
SecurityGroupRuleService	SecurityGroupRule create(SecurityGroupRule rule)	POST /v2.0/security-group-rules **Link**
	void delete(String id)	DELETE /v2.0/security-group-rules/{security-group-rules-id} **Link**
	SecurityGroupRule get(String id)	GET /v2.0/security-group-rules/{security-group-rules-id} **Link**
	List<? extends SecurityGroupRule> list()	GET /v2.0/security-group-rules **Link**
SecurityGroupService	SecurityGroup create(SecurityGroup securityGroup)	POST /v2.0/security-groups **Link**
	ActionResponse delete(String id)	DELETE /v2.0/security-groups/{security-group-id} **Link**

Interface	Method	API
	SecurityGroup get(String id)	GET /v2.0/security-groups/{security-group-id} **Link**
	List<? extends SecurityGroup>list()	GET /v2.0/security-groups **Link**
SubnetService	Subnet create(Subnet subnet)	POST /v2.0/subnets **Link**
	ActionResponse delete(String subnetId)	DELETE /v2.0/subnets/{subnet_id} **Link**
	Subnet get(String subnetId)	GET /v2.0/subnets/{subnet_id} **Link**
	List<? extends Subnet>list()	GET /v2.0/subnets **Link**
	List<? extends Subnet>list(Map<String,String> filteringParams)	GET /v2.0/subnets **Link**
	Subnet update(String subnetId, Subnet subnet)	PUT /v2.0/subnets/{subnet_id} **Link**
	Subnet update(Subnet subnet)	PUT /v2.0/subnets/{subnet_id} **Link**

6.1.4 ECS

The SDK interfaces based on the Nova v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
ComputeFloatingIPService	ActionResponse addFloatingIP(Server server, String ipAddress)	POST /v2/{project_id}/servers/{server_id}/action **Link**

Interface	Method	API
	ActionResponse addFloatingIP(Server server, String fixedIpAddress, String ipAddress)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	FloatingIP allocateIP(String pool)	POST /v2/{project_id}/os-floating-ips **Link**
	ActionResponse deallocateIP(String id)	DELETE /v2/{project_id}/os-floating-ips/{floating_ip_id} **Link**
	List<? extends FloatingIP> list()	GET /v2/{project_id}/os-floating-ips **Link**
	ActionResponse removeFloatingIP(Server server, String ipAddress)	POST /v2/{project_id}/servers/{server_id}/action **Link**
ComputeImageService	ActionResponse delete(String imageId)	DELETE /v2/{project_id}/images/{image_id} **Link**
	Image get(String imageId)	GET /v2/{project_id}/images/{image_id} **Link**
	List<? extends Image> list()	GET /v2/{project_id}/images/detail **Link**
	List<? extends Image> list(boolean detailed)	• detailed=true: GET /v2/{project_id}/images/detail **Link** • detailed=false: GET /v2/{project_id}/images **Link**
ComputeSecurityGroupService	SecGroupExtension create(String name, String description)	POST /v2/{project_id}/os-security-groups **Link**

Interface	Method	API
	SecGroupExtension.Rule createRule(SecGroupExtension.Rule rule)	POST /v2/{project_id}/os-security-group-rules **Link**
	ActionResponse delete(String securityGroupId)	DELETE /v2/{project_id}/os-security-groups/{security_group} **Link**
	ActionResponse deleteRule(String ruleId)	DELETE /v2/{project_id}/os-security-group-rules/{security_group_rule_id} **Link**
	SecGroupExtension get(String securityGroupId)	GET /v2/{project_id}/os-security-groups/{security_group_id} **Link**
	List<? extends SecGroupExtension> list()	GET /v2/{project_id}/os-security-groups **Link**
	SecGroupExtension update(String securityGroupId, String name, String description)	PUT /v2/{project_id}/os-security-groups/{security_group_id} **Link**
FlavorService	Flavor get(String flavorId)	GET /v2/{project_id}/flavors/{flavor_id} **Link**
	List<? extends Flavor> list()	GET /v2/{project_id}/flavors/detail **Link**
KeypairService	Keypair create(String name, String publicKey)	POST /v2/{project_id}/os-keypairs **Link**
	ActionResponse delete(String name)	DELETE /v2/{project_id}/os-keypairs/{keypair_name} **Link**

Interface	Method	API
	Keypair get(String name)	GET /v2/{project_id}/os-keypairs/{keypair_name} Link
	List<? extends Keypair> list()	GET /v2/{project_id}/os-keypairs Link
InstanceActions-Service	List<? extends InstanceAction> list(String serverId)	GET /v2/{project_id}/servers/{server_id}/os-instance-actions Link
	InstanceAction get(String serverId, String requestId)	GET /v2/{project_id}/servers/{server_id}/os-instance-actions/{request_id} Link
QuotaSetService	QuotaSet get(String tenantId)	GET /v2/{project_id}/os-quota-sets/{project_id} Link
	QuotaSet get(String tenantId, String userId)	GET /v2/{project_id}/os-quota-sets/{project_id}?user_id={user_id} Link
	Limits limits()	GET /v2/{project_id}/limits Link
	getDefault(String tenantId)	GET /v2/{project_id}/os-quota-sets/{project_id}/defaults Link
ServerGroupService	ServerGroup create(String name, String policy)	POST /v2/{project_id}/os-server-groups Link
	ActionResponse delete(String id)	DELETE /v2/{project_id}/os-server-groups/{server_group_id} Link

Interface	Method	API
	ServerGroup get(String id)	GET /v2/{project_id}/os-server-groups/{server_group_id} **Link**
	List<? extends ServerGroup> list()	GET /v2/{project_id}/os-server-groups **Link**
ServerService	VolumeAttachment getAttachVolume(String serverId, String volumeId)	GET /v2/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id} **Link**
	List<? extends VolumeAttachment> listAttachedVolumes(String serverId)	GET /v2/{project_id}/servers/{server_id}/os-volume_attachments **Link**
	ActionResponse action(String serverId, Action action) Executes the specified Action such as RESUME, PAUSE, START, and STOP.	POST /v2/{project_id}/servers/{server_id}/action **Link** (START) **Link** (STOP)
	ActionResponse stop(String serverId, StopType type)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	VolumeAttachment attachVolume(String serverId, String volumeId, String device)	POST /v2/{project_id}/servers/{server_id}/os-volume_attachments **Link**
	Server boot(ServerCreate server)	POST /v2/{project_id}/servers **Link**
	Server bootAndWaitActive(ServerCreate server, int maxWaitTime)	POST /v2/{project_id}/servers **Link** GET /v2/{project_id}/servers/{server_id} **Link**

Interface	Method	API
	ActionResponse confirmResize(String serverId)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	String createSnapshot(String serverId, String snapshotName)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	ActionResponse delete(String serverId)	DELETE /v2/{project_id}/servers/{server_id} **Link**
	ActionResponse deleteMetadataItem(String serverId, String key)	DELETE /v2/{project_id}/servers/{server_id}/metadata/{key} **Link**
	ActionResponse detachVolume(String serverId, String attachmentId)	DELETE /v2/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id} **Link**
	ActionResponse detachVolume(String serverId, String volumeId, int deleteFlag)	DELETE /v2/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id}?delete_flag={delete_flag} **Link**
	Server get(String serverId)	GET /v2/{project_id}/servers/{server_id} **Link**
	Map<String,String> getMetadata(String serverId)	GET /v2/{project_id}/servers/{server_id}/metadata **Link**
	List<? extends Server> list()	GET /v2/{project_id}/servers/detail **Link**

Interface	Method	API
	List<? extends Server> list(boolean detail)	• detail=true: GET /v2/{project_id}/servers/detail **Link** • detail=false: GET /v2/{project_id}/servers **Link**
	List<? extends Server> list(Map<String,String> filteringParams)	GET /v2/{project_id}/servers/detail{?changes-since,image,flavor,name,status,host,limit,marker} **Link**
	ActionResponse reboot(String serverId, RebootType type)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	ActionResponse resize(String serverId, String flavorId)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	ActionResponse revertResize(String serverId)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	Server update(String serverId, ServerUpdateOptions options)	PUT /v2/{project_id}/servers/{server_id} **Link**
	Map<String,String> updateMetadata(String serverId, Map<String,String> metadata)	PUT /v2/{project_id}/servers/{server_id}/metadata **Link**
	Server waitForServerStatus(String serverId, Server.Status status, int maxWait, TimeUnit maxWaitUnit)	GET /v2/{project_id}/servers/{server_id} **Link**

Interface	Method	API
	List<? extends Server> list(boolean detail , Map<String, String> filteringParams)	• detail = true: GET /v2/{project_id}/servers/detail{?changes-since,image,flavor,name,status,limit,marker,not-tags,reservation_id,ip} **Link** • detail = false: GET /v2/{project_id}/servers{?changes-since,image,flavor,name,status,limit,marker,not-tags,reservation_id} **Link**
	String getConsoleOutput(String serverId, int numLines)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	Map<String, String> getMetadataItem(String serverId, String key)	GET /v2/{project_id}/servers/{server_id}/metadata/{key} **Link**
	Map<String, String> setMetadataItem(String serverId, String key, String value)	PUT /v2/{project_id}/servers/{server_id}/metadata/{key} **Link**
InterfaceService (ext)	InterfaceAttachment create(String serverId, String portId)	POST /v2/{project_id}/servers/{server_id}/os-interface **Link**
	InterfaceAttachment create(String serverId, NovaInterfaceAttachmentCreate novaInterfaceAttachmentCreate)	POST /v2/{project_id}/servers/{server_id}/os-interface **Link**
	ActionResponse detach(String serverId, String attachmentId)	DELETE /v2/{project_id}/servers/{server_id}/os-interface/{port_id} **Link**
	InterfaceAttachment get(String serverId, String attachmentId)	GET /v2/{project_id}/servers/{server_id}/os-interface/{port_id} **Link**

Interface	Method	API
	List<? extends InterfaceAttachment> list(String serverId)	GET /v2/{project_id}/servers/{server_id}/os-interface **Link**
ZoneService(ext)	List<? extends AvailabilityZone> list()	GET /v2/{project_id}/os-availability-zone **Link**

The SDK interfaces based on the Nova v2.1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
ServerService	NovaRemoteConsoleResponse getRemoteConsole(String serverId, NovaRemoteConsole remoteConsole, Map<String, String> headers)	POST /v2.1/{project_id}/servers/{server_id}/remote-consoles **Link**
ServerTagService	NovaServerTag list(String serverId)	GET /v2.1/{project_id}/servers/{server_id}/tags **Link**
	NovaServerTag addTags(String serverId, NovaServerTag tags)	PUT /v2.1/{project_id}/servers/{server_id}/tags **Link**
	ActionResponse delete(String serverId, String tag)	DELETE /v2.1/{project_id}/servers/{server_id}/tags **Link**
	ActionResponse check(String serverId, String tag)	GET /v2.1/{project_id}/servers/{server_id}/tags/{tag} **Link**
	ActionResponse addSingle(String serverId, String tag)	PUT /v2.1/{project_id}/servers/{server_id}/tags/{tag} **Link**

Interface	Method	API
	ActionResponse deleteAll(String serverId)	DELETE /v2.1/{project_id}/servers/{server_id}/tags/{tag} **Link**

The SDK interfaces based on the ECS v1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
CloudServerService	String create(ServerCreate creation)	POST /v1/{project_id}/cloudservers **Link**
	List<CloudServer> list()	GET /v1/{project_id}/cloudservers/detail **Link**
	CloudServers listWithCount()	GET /v1/{project_id}/cloudservers/detail **Link**
	CloudServer get(String serverId)	GET /v1/{project_id}/cloudservers/{server_id} **Link**
	String resize(ResizeServer resize,String serverId)	POST /v1/{project_id}/cloudservers/{server_id}/resize **Link**
	String delete(List<String> serverIds, boolean deletePublicIp, boolean deleteVolume)	POST /v1/{project_id}/cloudservers/delete **Link**
	String stop(List<String> serverIds, StopType type)	POST /v1/{project_id}/cloudservers/action **Link**
	String reboot(List<String> serverIds, RebootType type)	POST /v1/{project_id}/cloudservers/action **Link**
	String start(List<String> serverIds)	POST /v1/{project_id}/cloudservers/action **Link**

Interface	Method	API
	List<CloudServer> list(Map<String, String> filteringParams)	GET /v1/{project_id}/cloudservers/detail{?flavor,name,status,limit,offset,not-tags,reservation_id,enterprise_project_id} **Link**
	CloudServers listWithCount(Map<String, String> filteringParams)	GET /v1/{project_id}/cloudservers/detail{?flavor,name,status,limit,offset,not-tags,reservation_id,enterprise_project_id} **Link**
	List<Flavor> getSpecifications(String availabilityZone)	GET /v1/{project_id}/cloudservers/flavors{?availability_zone} **Link**
	CloudAbsoluteLimit limits()	GET /v1/{project_id}/cloudservers/limits **Link**
	SupportAutoRecovery getAutoRecovery(String serverId)	GET /v1/{project_id}/cloudservers/{server_id}/autorecovery **Link**
	ActionResponse manageAutoRecovery(String serverId, SupportAutoRecovery supportAutoRecovery)	PUT /v1/{project_id}/cloudservers/{server_id}/autorecovery **Link**
	RemoteConsoleResponse remoteConsole(String serverId, RemoteConsole remoteConsole)	POST /v1/{project_id}/cloudservers/{server_id}/remote_console **Link**
JobService	Job get(String jobId)	GET /v1/{project_id}/jobs/{job_id} **Link**
InterfaceService	InterfaceAttachments list(String serverId)	GET /v1/{project_id}/cloudservers/{server_id}/os-interface **Link**

Interface	Method	API
TagService	ActionResponse add(String serverId, List<ServerTags> serverTags)	POST /v1/{project_id}/cloudservers/{server_id}/tags/action **Link**
	ActionResponse delete(String serverId, List<ServerTags> serverTags)	POST /v1/{project_id}/cloudservers/{server_id}/tags/action **Link**
	CloudServerTag list(String serverId)	GET /v1/{project_id}/cloudservers/{server_id}/tags **Link**
	ProjectTag listProjectTags()	GET /v1/{project_id}/cloudservers/tags **Link**

The SDK interfaces based on the ECS v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
CloudServerV2Service	AsyncJobEntity reinstallOS(OSReinstallosReinstall, String serverId)	POST /v2/{project_id}/cloudservers/{server_id}/reinstallos **Link**

The SDK interfaces based on the ECS v1.1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
CloudServerService	AsyncServerRespEntity create(ServerCreate creation)	POST /v1.1/{project_id}/cloudservers **Link**
	AsyncRespEntity resize(ResizeServer resize,String serverId)	POST /v1.1/{project_id}/cloudservers/{server_id}/resize **Link**

6.1.5 EVS

The SDK interfaces based on the EVS v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
VolumeService	String create(CloudVolumes volume)	POST /v2/{project_id}/cloudvolumes **Link**
	String extend(Extend extend,String volumeId)	POST /v2/{project_id}/cloudvolumes/{volume_id}/action **Link**
	CloudVolumeResponse list()	GET /v2/{project_id}/cloudvolumes/detail **Link**
	CloudVolumesResponse list(Map<String,Object> filteringParams)	GET /v2/{project_id}/cloudvolumes/detail **Link**
VolumeSnapshotService	CloudVolumeSnapshotsResponse list()	GET /v2/{project_id}/cloudsnapshots/detail **Link**
	CloudVolumeSnapshotsResponse list(Map<String,Object> filteringParams)	GET /v2/{project_id}/cloudsnapshots/detail **Link**
	RollbackResponse rollback(String snapshotId)	POST /v2/{project_id}/cloudsnapshots/{snapshot_id}/rollback **Link**
	RollbackResponse rollback(String snapshotId, Rollback rollback)	POST /v2/{project_id}/cloudsnapshots/{snapshot_id}/rollback **Link**
JobService	Job get(String job_id)	GET /v1/{project_id}/jobs/{job_id} **Link**

The SDK interfaces based on the EVS v2.1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
VolumeService	AsyncRespEntity create(Volumes volume)	POST /v2.1/{project_id}/cloudvolumes **Link**
	AsyncRespEntity extend(Extend extend,String volumeId)	POST /v2.1/{project_id}/cloudvolumes/{volume_id}/action **Link**

The SDK interfaces based on the Cinder v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
BlockVolumeService	Volume create(Volume volume)	POST /v2/{project_id}/volumes **Link**
	Volume create(Volume volume, Map<String, Object> schedulerHints)	POST /v2/{project_id}/volumes **Link**
	Volume update(String volumeId, CinderVolumeUpdate volume)	PUT /v2/{project_id}/volumes/{volume_id} **Link**
	ActionResponse delete(String volumeId)	DELETE /v2/{project_id}/volumes/{volume_id} **Link**
	ActionResponse extend(String volumeId, Integer newSize)	POST /v2/{project_id}/volumes/{volume_id}/action **Link**
	Volume get(String volumeId)	GET /v2/{project_id}/volumes/{volume_id} **Link**
	List<? extends Volume> list()	GET /v2/{project_id}/volumes/detail **Link**

Interface	Method	API
	List<? extends Volume> list(Map<String,String> filteringParams)	• GET /v2/{project_id}/volumes/detail?limit={limit_nmuber} • GET /v2/{project_id}/volumes/detail?marker={volume_id} **Link**
	ActionResponse update(String volumeId, String name, String description)	PUT /v2/{project_id}/volumes/{volume_id} **Link**
	VolumeUploadImage uploadToImage(String volumeId, UploadImageData data)	POST /v2/{project_id}/volumes/{volume_id}/action **Link**
	ActionResponse readOnlyModeUpdate(String volumeId, boolean readonly)	POST /v2/{project_id}/volumes/{volume_id}/action **Link**
	List<? extends VolumeType> listVolumeTypes()	GET /v2/{project_id}/types **Link**
BlockVolumeSnapshotService	ActionResponse delete(String snapshotId)	DELETE /v2/{project_id}/snapshots/{snapshot_id} **Link**
	VolumeSnapshot create(VolumeSnapshot snapshot)	POST /v2/{project_id}/snapshots **Link**
	ActionResponse update(String snapshotId, String name, String description)	PUT /v2/{project_id}/snapshots/{snapshot_id} **Link**
	VolumeSnapshot get(String snapshotId)	GET /v2/{project_id}/snapshots/{snapshot_id} **Link**

Interface	Method	API
	List<? extends VolumeSnapshot> list()	GET /v2/{project_id}/snapshots **Link**
	List<? extends VolumeSnapshot> list(Map<String,String> filteringParams)	GET /v2/{project_id}/snapshots?volume_id={volume_id} **Link**
CinderZoneService	List<? extends AvailabilityZone> list()	GET /v2/{project_id}/os-availability-zone **Link**
BlockVolumeTransferService	List<? extends VolumeTransfer> list()	GET /v2/{project_id}/os-volume-transfer/detail **Link**
	List<? extends VolumeTransfer> list(boolean detailed)	• GET /v2/{project_id}/os-volume-transfer **Link** • GET /v2/{project_id}/os-volume-transfer/detail **Link**
	VolumeTransfer get(String transferId)	GET /v2/{project_id}/os-volume-transfer/{transfer_id} **Link**
	VolumeTransfer create(String volumeId)	POST /v2/{project_id}/os-volume-transfer **Link**
	VolumeTransfer create(String volumeId, String name)	POST /v2/{project_id}/os-volume-transfer **Link**
	VolumeTransfer accept(String transferId, String authKey)	POST /v2/{project_id}/os-volume-transfer/{transfer_id}/accept **Link**

Interface	Method	API
	ActionResponse delete(String transferId)	DELETE /v2/{project_id}/os-volume-transfer/{transfer_id} **Link**

6.1.6 AS

The SDK interfaces based on the AS v1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
AutoScalingGroup Servic	String create(ScalingGroupCreate group)	POST /autoscaling-api/v1/{project_id}/scaling_group **Link**
	List<? extends ScalingGroup> list()	GET /autoscaling-api/v1/{project_id}/scaling_group **Link**
	ScalingGroup get(String groupId)	GET /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id} **Link**
	String update(String groupId, ScalingGroupUpdate group)	PUT /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id} **Link**
	ActionResponse delete(String groupId)	DELETE /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id **Link**
	ActionResponse resume(String groupId)	POST /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id}/action **Link**
	ActionResponse pause(String groupId)	POST /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id}/action **Link**

Interface	Method	API
AutoScalingConfigService	String create(ScalingConfigCreate config)	POST /autoscaling-api/v1/{project_id}/scaling_configuration **Link**
	List<? extends ScalingConfig> list()	GET /autoscaling-api/v1/{project_id}/scaling_configuration **Link**
	ScalingConfig get(String configId)	GET /autoscaling-api/v1/{project_id}/scaling_configuration/{scaling_configuration_id} **Link**
	ActionResponse delete(String configId)	DELETE /autoscaling-api/v1/{project_id}/scaling_configuration/{scaling_configuration_id} **Link**
	ActionResponse delete(List<String> configIds)	POST /autoscaling-api/v1/{project_id}/scaling_configurations **Link**
AutoScalingGroupInstanceService	List<? extends ScalingGroupInstance> list(String groupId, ScalingGroupInstanceListOptions options)	GET /autoscaling-api/v1/{project_id}/scaling_group_instance/{scaling_group_id}/list **Link**
	ActionResponse delete(String instanceId, boolean deleteInstance)	DELETE /autoscaling-api/v1/{project_id}/scaling_group_instance/{instance_id} **Link**
	ActionResponse batchAdd(String groupId, List<String> instanceIds, boolean deleteInstance)	POST /autoscaling-api/v1/{project_id}/scaling_group_instance/{scaling_group_id}/action **Link**

Interface	Method	API
	ActionResponse batchRemove(String groupId, List<String> instanceIds, boolean deleteInstance)	POST /autoscaling-api/v1/{project_id}/scaling_group_instance/{scaling_group_id}/action **Link**
AutoScalingPolicyService	String create(ScalingPolicyCreateUpdate policy)	POST /autoscaling-api/v1/{project_id}/scaling_policy **Link**
	String update(ScalingPolicyCreateUpdate policy)	PUT /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**
	List<? extends ScalingPolicy> list(String groupId)	GET /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_group_id}/list **Link**
	ScalingPolicy get(String policyId)	GET /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**
	ActionResponse execute(String policyId)	POST /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id}/action **Link**
	ActionResponse resume(String policyId)	POST /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id}/action **Link**
	ActionResponse pause(String policyId)	POST /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id}/action **Link**
	ActionResponse delete(String policyId)	DELETE /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**
AutoScalingActivityLogService	list(String groupId, ScalingActivityLogListOptions options)	GET /autoscaling-api/v1/{project_id}/scaling_activity_log/{scaling_group_id} **Link**

Interface	Method	API
AutoScalingQuotaService	List<Quota> list()	GET /autoscaling-api/v1/{project_id}/quotas **Link**
	List<Quota> list(String groupId)	GET /autoscaling-api/v1/{project_id}/quotas/{scaling_group_id} **Link**
AutoScalingLifecycleHookService	ASAutoScalingLifecycleHook create(ASAutoScalingLifecycle-Hook lifecycleHook, String groupId)	POST /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id} **Link**
	List<? extends ASAutoScalingLife-cycleHook> list(String groupId)	GET /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/list **Link**
	ASAutoScalingLifecycleHook list(String groupId, String lifecycleHookName)	GET /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
	ActionResponse delete(String groupId, String lifecycleHookName)	DELETE /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
	ASAutoScalingLifecycleHook update(String groupId, String lifecycleHookName, ASAutoScalingLifecycleHook lifecycleHook)	PUT /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
	List<? extends AutoScalingInstan-ceHangupInfo> scalingInstance-Hangup(String groupId, ScalingInstanceOptions instanceId)	GET /autoscaling-api/v1/{project_id}/scaling_instance_hook/{scaling_group_id}/list **Link**

Interface	Method	API
	ActionResponse scalingInstanceHookCallback(String groupId, ASAutoScalingLifecycleInstanceCallback lifecycleInstanceCallback)	PUT /autoscaling-api/v1/{project_id}/scaling_instance_hook/{scaling_group_id}/callback **Link**
AutoScalingInformService	ASAutoScalingInform deploy(String groupId, ASAutoScalingInform info)	PUT /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id} **Link**
	ASAutoScalingTopics list(String groupId)	GET /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id} **Link**
	ActionResponse delete(String groupId, String topicUrn)	DELETE /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id}/{topic_urn} **Link**

6.1.7 Cloud Eye

The SDK interfaces based on the Cloud Eye v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
MetricService	List<? extends Metric> getList(MetricFilterOptions options);	GET /V1.0/{project_id}/metrics
AlarmService	List<? extends Alarm> list(AlarmFilterOptions options);	GET /V1.0/{project_id}/alarms
	List<? extends Alarm> get(String alarmId);	GET /V1.0/{project_id}/alarms/{alarm_id}
	ActionResponse startAlarm(String alarmId)	PUT /V1.0/{project_id}/alarms/{alarm_id}/action
	ActionResponse stopAlarm(String alarmId)	PUT /V1.0/{project_id}/alarms/{alarm_id}/action

Interface	Method	API
	ActionResponse deleteAlarm(String alarmId);	DELETE /V1.0/{project_id}/alarms/{alarm_id}
MetricDataService	MetricAggregation get(String namespace, String metric_name, Date from, Date to, Period period, Filter filter, String[] dimValues);	GET /V1.0/{project_id}/metric-data
	ActionResponse add(List<? extends MetricData> metrics);	POST /V1.0/{project_id}/metric-data
QuotaService	CloudEyeQuota get();	GET /V1.0/{project_id}/quotas

6.1.8 DNS

The SDK interfaces based on the DNS v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
ZoneService	Zone create(Zone zone)	POST /v2/zones **Link** (public network) **Link** (intranet)
	Zone get(String zoneId)	GET /v2/zones/{zone_id} **Link** (public network) **Link** (intranet)
	List<? extends Zone> list()	GET /v2/zones **Link** (public network) **Link** (intranet)
	Zone delete(String zoneId)	DELETE /v2/zones/{zone_id} **Link** (public network) **Link** (intranet)
	List<? extends Nameserver> listNameservers(String zoneId)	GET /v2/zones/{zone_id}/nameservers **Link** (public network) **Link** (intranet)
	DesignateZone.Router associateRouter(String zoneId, DesignateZone.Router router)	POST /v2/zones/{zone_id}/associaterouter **Link**

Interface	Method	API
	DesignateZone.Router disassociateRouter(String zoneId, DesignateZone.Router router)	POST /v2/zones/{zone_id}/disassociaterouter **Link**
RecordsetService	Recordset create(String zoneId, Recordset recordSet)	POST /v2/zones/{zone_id}/recordsets **Link**
	Recordset get(String zoneId, String recordsetId)	GET /v2/zone/{zone_id}/recordsets/{recordset_id} **Link**
	List<? extends Recordset> list()	GET /v2/recordsets **Link**
	List<? extends Recordset> list(RecordsetListOptions options)	GET /v2/zones/{zone_id}/recordsets **Link**
	Recordset delete(String zoneId, String recordsetId)	DELETE /v2/zones/{zone_id}/recordsets/{recordset_id} **Link**
PTRService	DesignatePTR setup(DesignatePTR record)	PATCH /v2/reverse/floatingips/{region}:{floatingip_id} **Link**
	ActionResponse restore(String region, String floatingIpId)	PATCH /v2/reverse/floatingips/{region}:{floatingip_id} **Link**
	List<? extends PTR> list()	GET /v2/reverse/floatingips **Link**
	DesignatePTR get(String region, String floatingIpId)	GET /v2/reverse/floatingips/{region}:{floatingip_id} **Link**

6.1.9 ELB

The SDK interfaces based on the ELB v2.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
LbRuleV2Service	NeutronRules list(String l7policyId)	GET /v2.0/lbaas/l7policies/{l7policy_id}/rules **Link**
	NeutronRule get(String l7policyId, String ruleId)	GET /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} **Link**
	NeutronRule create(NeutronRule ruleModel, String l7policyId)	POST /v2.0/lbaas/l7policies/{l7policy_id}/rules **Link**
	NeutronRule update(NeutronRuleUpdate updateModel, String l7policyId, String ruleId)	PUT /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} **Link**
	ActionResponse delete(String l7policyId, String ruleId)	DELETE /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} **Link**
LbWhitelistV2Service	NeutronWhitelists list()	GET /v2.0/lbaas/whitelists **Link**
	NeutronWhitelist get(String whitelistId)	GET /v2.0/lbaas/whitelists/{whitelist_id} **Link**
	NeutronWhitelist create(NeutronWhitelist model)	POST /v2.0/lbaas/whitelists **Link**
	NeutronWhitelist update(NeutronWhitelist model,String whitelistId)	PUT /v2.0/lbaas/whitelists/{whitelist_id} **Link**
	ActionResponse delete(String whitelistId)	DELETE /v2.0/lbaas/whitelists/{whitelist_id} **Link**
LbCertificateV2Service	NeutronCertificates list()	GET /v2.0/lbaas/certificates **Link**
	NeutronCertificate get(String id)	GET /v2.0/lbaas/certificates/{certificate_id} **Link**
	NeutronCertificate create(NeutronCertificate model)	POST /v2.0/lbaas/certificates **Link**

Interface	Method	API
	NeutronCertificate update(NeutronCertificateUpdate model, String id)	PUT /v2.0/lbaas/certificates/{certificate_id} **Link**
	ActionResponse delete(String id)	DELETE /v2.0/lbaas/certificates/{certificate_id} **Link**
LbPolicyV2Service	NeutronL7Policies list()	GET /v2.0/lbaas/l7policies **Link**
	NeutronL7Policy get(String policyId)	GET /v2.0/lbaas/l7policies/{policy_id} **Link**
	NeutronL7Policy create(NeutronL7Policy model)	POST /v2.0/lbaas/l7policies **Link**
	NeutronL7Policy update(NeutronL7PolicyUpdate l7PolicyUpdate, String l7policyId)	PUT /v2.0/lbaas/l7policies/{policy_id} **Link**
	ActionResponse delete(String l7policyId)	DELETE /v2.0/lbaas/l7policies/{policy_id} **Link**
LoadBalancerV2Service	List<? extends LoadBalancerV2> list()	GET /v2.0/lbaas/loadbalancers **Link**
	LoadBalancerV2 get(String loadbalancerId)	GET /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	LoadBalancerV2 create(LoadBalancerV2 loadbalancer)	POST /v2.0/lbaas/loadbalancers **Link**
	LoadBalancerV2 update(String loadbalancerId, LoadBalancerV2Update loadbalancer)	PUT /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	ActionResponse delete(String loadbalancerId)	DELETE /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	LoadBalancerV2StatusTree statusTree(String loadbalancerId)	GET /v2.0/lbaas/loadbalancers/{loadbalancer_id}/statuses **Link**

Interface	Method	API
ListenerV2Service	List<? extends ListenerV2> list()	GET /v2.0/lbaas/listeners **Link**
	ListenerV2 get(String listenerId)	GET /v2.0/lbaas/listeners/{listener_id} **Link**
	ListenerV2 create(ListenerV2 listener)	POST /v2.0/lbaas/listeners **Link**
	ListenerV2 update(String listenerId, ListenerV2Update listener)	PUT /v2.0/lbaas/listeners/{listener_id} **Link**
	ActionResponse delete(String listenerId)	DELETE /v2.0/lbaas/listeners/{listener_id} **Link**
LbPoolV2Service	List<? extends LbPoolV2> list()	GET /v2.0/lbaas/pools **Link**
	LbPoolV2 get(String lbPoolId)	GET /v2.0/lbaas/pools/{pool_id} **Link**
	LbPoolV2 create(LbPoolV2 lbPool)	POST /v2.0/lbaas/pools **Link**
	LbPoolV2 update(String lbPoolId, LbPoolV2Update lbPool)	PUT /v2.0/lbaas/pools/{pool_id} **Link**
	ActionResponse delete(String lbPoolId)	DELETE /v2.0/lbaas/pools/{pool_id} **Link**
HealthMonitorV2Service	List<? extends HealthMonitorV2> list()	GET /v2.0/lbaas/healthmonitors **Link**
	HealthMonitorV2 get(String healthMonitorId)	GET /v2.0/lbaas/healthmonitors/{healthmonitor_id} **Link**
	HealthMonitorV2 create(HealthMonitorV2 healthMonitor)	POST /v2.0/lbaas/healthmonitors **Link**
	HealthMonitorV2 update(String healthMonitorId, HealthMonitorV2Update healthMonitor)	PUT /v2.0/lbaas/healthmonitors/{healthmonitor_id} **Link**

Interface	Method	API
	ActionResponse delete(String healthMonitorId)	DELETE /v2.0/lbaas/healthmonitors/{healthmonitor_id} **Link**
LbPoolV2Service	List<? extends MemberV2> listMembers(String lbPoolId)	GET /v2.0/lbaas/pools/{pool_id}/members **Link**
	MemberV2 getMember(String lbPoolId, String memberId)	GET /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**
	MemberV2 createMember(String lbPoolId, MemberV2 member)	POST /v2.0/lbaas/pools/{pool_id}/members **Link**
	MemberV2 updateMember(String lbPoolId, String memberId, MemberV2Update member)	PUT /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**
	ActionResponse deleteMember(String lbPoolId, String memberId)	DELETE /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**

The SDK interfaces based on the ELB v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
ELBLoadBalancerService	ELBJob create(LoadBalancerCreate loadBalancer)	POST /v1.0/{project_id}/elbaas/loadbalancers **Link**
	LoadBalancer get(String loadBalancerId)	GET /v1.0/{project_id}/elbaas/loadbalancers/{loadbalancer_id} **Link**
	List<? extends LoadBalancer> list()	GET /v1.0/{project_id}/elbaas/loadbalancers **Link**

Interface	Method	API
	ELBJob update(String loadBalancerId, LoadBalancerUpdate loadBalancer)	PUT /v1.0/{project_id}/elbaas/loadbalancers/{loadbalancer_id} **Link**
	ELBJob delete(String loadBalancerId)	DELETE /v1.0/{project_id}/elbaas/loadbalancers/{loadbalancer_id} **Link**
ELBListenerService	ListenerCreate create(ListenerCreate listener)	POST /v1.0/{project_id}/elbaas/listeners **Link**
	Listener get(String listenerId)	GET /v1.0/{project_id}/elbaas/listeners/{listener_id} **Link**
	Listener[] list()	GET /v1.0/{project_id}/elbaas/listeners?loadbalancer_id={loadbalancer_id} **Link**
	Listener update(String listenerId, ListenerUpdate listener)	PUT /v1.0/{project_id}/elbaas/listeners/{listener_id} **Link**
	ActionResponse delete(String listenerId)	DELETE /v1.0/{project_id}/elbaas/listeners/{listener_id} **Link**
ELBHealthCheckService	HealthCheck create(HealthCheckCreate healthCheck)	POST /v1.0/{project_id}/elbaas/healthcheck **Link**
	HealthCheck get(String healthCheckId)	GET /v1.0/{project_id}/elbaas/healthcheck/{healthcheck_id} **Link**
	HealthCheck update(String healthCheckId, HealthCheckUpdate healthCheck)	PUT /v1.0/{project_id}/elbaas/healthcheck/{healthcheck_id} **Link**
	ActionResponse delete(String healthCheckId)	DELETE /v1.0/{project_id}/elbaas/healthcheck/{healthcheck_id} **Link**

Interface	Method	API
ELBServerService	ELBJob create(String listenerId, List<ServerCreate> servers)	POST /v1.0/{project_id}/elbaas/listeners/{listener_id}/members **Link**
	ELBJob delete(String listenerId, ServerDelete serverDelete)	POST /v1.0/{project_id}/elbaas/listeners/{listener_id}/members/action **Link**
	Server[] list(String listenerId)	GET /v1.0/{project_id}/elbaas/listeners/{listener_id}/members **Link**
ELBCertificateService	Certificate create(Certificate cert)	POST /v1.0/{project_id}/elbaas/certificate **Link**
	Certificates list()	GET /v1.0/{project_id}/elbaas/certificate **Link**
	Certificate update(String certificateId, CertificateUpdate cert)	PUT /v1.0/{project_id}/elbaas/certificate/{certificate_id} **Link**
	ActionResponse delete(String certificateId)	DELETE /v1.0/{project_id}/elbaas/certificate/{certificate_id} **Link**

6.1.10 VBS

The SDK interfaces based on the VBS v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
AsyncVolumeBackupService	AsyncVolumeBackupJob create(AsyncVolumeBackupCreate cvbc)	POST /v2/{tenant_id}/cloudbackups
	VolumeBackup create(VolumeBackupCreate vbc)	Post /v2/{project_id}/backups

Interface	Method	API
	AsyncVolumeBackupJob restore(String volumeBackupId, String volumeId)	POST /v2/{tenant_id}/cloudbackups/{backup_id}/restore
	List<? extends VolumeBackup> list(boolean detail, Map<String, String> filteringParams)	GET /v2/{tenant_id}/backups
	List<? extends VolumeBackup> list(boolean detail, Map<String, String> filteringParams)	GET /v2/{tenant_id}/backups/detail
	VolumeBackup get(String backupId)	GET /v2/{tenant_id}/backups/{backup_id}
	ActionResponse delete(String backupId)	DELETE /v2/{tenant_id}/backups/{backup_id}
	AsyncVolumeBackupJob get(String jobId)	GET /v1/{tenant_id}/jobs/{job_id}
BlockVolumeBackupPolicyService	VolumeBackupPolicy create(VolumeBackupPolicy policy)	POST /v2/{tenant_id}/backuppolicy
	List<? extends VolumeBackupPolicy> list()	GET /v2/{tenant_id}/backuppolicy
	VolumeBackupPolicy update(VolumeBackupPolicy updated)	PUT /v2/{tenant_id}/backuppolicy/{policy_id}
	ActionResponse delete(String backupPolicyId)	DELETE /v2/{tenant_id}/backuppolicy/{policy_id}
	VolumeBackupPolicyResourceActionResult linkResources(String backupPolicyId, List<String> resourceIds)	POST /v2/{tenant_id}/backuppolicyresources
	VolumeBackupPolicyResourceActionResult unlinkResources(String backupPolicyId, List<String> resourceIds)	POST /v2/{tenant_id}/backuppolicyresources/{policy_id}/deleted_resources
	ActionResponse execute(String backupPolicyId)	POST /v2/{tenant_id}/backuppolicy/{policy_id}/action

Interface	Method	API
	List<? extends VolumeBackupPolicyBackup-Task> tasks(String policyId, BakcupTaskListOptions options)	GET /v2/{tenant_id}/backuppolicy/{policy_id}/backuptasks
	VolumeBackupPolicy enable(String backupPolicyId)	PUT /v2/{tenant_id}/backuppolicy/{policy_id}
	VolumeBackupPolicy disable(String backupPolicyId)	PUT /v2/{tenant_id}/backuppolicy/{policy_id}

6.1.11 CTS

The SDK interfaces based on the CTS v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	method	API URL
TrackerService	Tracker create(String bucketName, String filePrefixName)	POST /v1.0/{project_id}/tracker
	Tracker get(String trackerName)	GET /v1.0/{project_id}/tracker{?tracker_name}
	Tracker update(TrackerUpdate update)	PUT /v1.0/{project_id}/tracker/{?tracker_name}
	ActionResponse delete(String trackerName)	DELETE /v1.0/{project_id}/tracker{?tracker_name}
TraceService	List<Trace> list(String trackerName, TraceListOptions options)	GET /v2.0/{project_id}/{tracker_name}/trace{?trace_id,service_type,resource_type,resource_id,resource_name,trace_name,trace_rating,user,limit,from,to,next}

6.1.12 KMS

The SDK interfaces based on the KMS v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
KeyService	Key create(KeyCreate keyCreate)	POST /v1.0/{project_id}/kms/create-key
	Keys list(KeyListOptions options)	POST /v1.0/{project_id}/kms/list-keys
	Key get(String keyId, String sequence)	POST /v1.0/{project_id}/kms/describe-key
	Key disable(String keyId, String sequence)	POST /v1.0/{project_id}/kms/disable-key
	Key enable(String keyId, String sequence)	POST /v1.0/{project_id}/kms/enable-key
	Key scheduleDeletion(String keyId, Integer pendingDays, String sequence)	POST /v1.0/{project_id}/kms/schedule-key-deletion
	Key cancelDeletion(String keyId, String sequence)	POST /v1.0/{project_id}/kms/cancel-key-deletion
	Integer getKeyCreatedAmount()	GET /v1.0/{project_id}/kms/user-instances
	List<Quota> quotas()	GET /v1.0/{project_id}/kms/user-quotas
CryptoService	String generateRandomString(String sequence)	POST /v1.0/{project_id}/kms/gen-random
	DEK createDEK(String keyId, HashMap<String, Object> encryptionContext, String sequence)	POST /v1.0/{project_id}/kms/create-datakey
	DEK createDEKWithoutPlaintext(String keyId, HashMap<String, Object> encryptionContext, String sequence)	POST /v1.0/{project_id}/kms/create-datakey-without-plaintext
	EncryptedDEK encryptDEK(EncryptDEK encrypt)	POST /v1.0/{project_id}/kms/encrypt-datakey

Interface	Method	API
	DecryptedDEK decryptDEK(DecryptDEK decrypt)	POST /v1.0/{project_id}/kms/decrypt-datakey

6.1.13 Anti-DDoS

The SDK interfaces based on Anti-DDoS v1 APIs are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
AntiDDoSService	AntiDDoSConfig listConfigs()	GET /v1/{project_id}/antiddos/query_config_list **Link**
	AntiDDoS get(String floatingIpId)	GET /v1/{project_id}/antiddos/{floating_ip_id} **Link**
	Task update(AntiDDoS entity, String floatingIpId)	PUT /v1/{project_id}/antiddos/{floating_ip_id} **Link**
	Task getTask(String taskId)	GET /v1/{project_id}/query_task_status **Link**
	AntiDDoSStatus listStatus()	GET /v1/{project_id}/antiddos **Link**
	AntiDDoSStatusDetail getStatus(String floatingIpId)	GET /v1/{project_id}/antiddos/{floating_ip_id}/status **Link**
	List<? extends AntiDDoSDailyData> dailyReport(String floatingIpId)	GET /v1/{project_id}/antiddos/{floating_ip_id}/daily **Link**
	List<? extends AntiDDoSLog> listLogs(String floatingIpId, AntiDDoSLogListOptions options)	GET /v1/{project_id}/antiddos/{floating_ip_id}/logs **Link**
	AntiDDoSWeeklyData weeklyReport(Date periodStartDate)	GET /v1/{project_id}/antiddos/weekly **Link**

Interface	Method	API
AntiDDoSWarnService	AntiDDoSWarn query()	GET /v2/{project_id}/ warnalert/alertconfig/query Link

6.1.14 DMS

The SDK interfaces based on the DMS v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
QueueService	Queue create(String name, String description)	POST /v1.0/{project_id}/ queues
	List<? extends Queue> list()	GET /v1.0/{project_id}/ queues
	Queue get(String queueId)	GET /v1.0/{project_id}/ queues/{queue_id}
	ActionResponse delete(String queueId)	DELETE /v1.0/{project_id}/ queues/{queue_id}
ConsumerGroupService	ConsumerGroup create(String queueId, String consumerGroupName)	POST /v1.0/{project_id}/ queues/{queue_id}/groups
	List<ConsumerGroup> list(String queueId)	GET /v1.0/{project_id}/ queues/{queue_id}/groups
	ActionResponse delete(String queueId, String consumerGroupId)	DELETE /v1.0/{project_id}/ queues/{queue_id}/groups/{consumer_group_id}
QueueMessageService	ActionResponse produce(String queueId, QueueMessage message)	POST /v1.0/{project_id}/ queues/{queue_id}/ messages
	List<QueueMessageWithHandler> consume(String queueId, String consumerGroupId, Integer maxMessages, Integer timeWait)	GET /v1.0/{project_id}/ queues/{queue_id}/groups/{consumer_group_id}/ messages

Interface	Method	API
	ConsumeConfirmResponse confirmConsuming(String queueId, String consumerGroupId,Map<String, ConsumeStatus> consumeResult)	POST /v1.0/{project_id}/queues/{queue_id}/groups/{consumer_group_id}/ack
MessageQueueQuotaService	List<Quota> get()	GET /v1.0/{project_id}/quotas/dms

6.1.15 MRS

The SDK interfaces based on the MRS v1.1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
ClusterService	MapReduceClusterCreateResult createAndRunJob(MapReduceClusterCreate cluster, MapReduceJobExeCreate jobExe)	POST /v1.1/{project_id}/run-job-flow
	ActionResponse expand(String clusterId, int amount)	PUT /v1.1/{project_id}/cluster_infos/{cluster_id}
	ActionResponse reduce(String clusterId, int amount, List<String> includes, List<String> excludes)	PUT /v1.1/{project_id}/cluster_infos/{cluster_id}
	MapReduceClusterInfo get(String clusterId)	GET /v1.1/{project_id}/cluster_infos/{cluster_id}
	ActionResponse delete(String clusterId)	DELETE /v1.1/{project_id}/clusters/{cluster_id}
JobExeServiceImpl	MapReduceJobExe create(MapReduceJobExeCreate jobExeCreate)	POST /v1.1/{project_id}/jobs/submit-job
	List<? extends MapReduceJobExe> list(JobExeListOptions options)	GET /v1.1/{project_id}/job-exes
	MapReduceJobExe get(String jobExeId)	GET /v1.1/{project_id}/job-exes/{job_exe_id}

Interface	Method	API
JobExecutionService	ActionResponse delete(String jobExecutionId)	DELETE /v1.1/{project_id}/job-executions/{job_execution_id}

6.1.16 CDN

The SDK interfaces based on the CDN v1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
DomainService	list(**params)	GET /v1.0/cdn/domains
	create(**attrs)	POST /v1.0/cdn/domains
	getDetail(domainId, **params)	GET /v1.0/cdn/domains/{domain_id}/detail
	delete(domainId, **params)	DELETE /v1.0/cdn/domains/{domain_id}
	enable(domainId, **params)	PUT /v1.0/cdn/domains/{domain_id}/enable
	disable(domainId, **params)	PUT /v1.0/cdn/domains/{domain_id}/disable
	setOrigin(domainId, **attrs, **params)	PUT /v1.0/cdn/domains/{domain_id}/origin
	setOriginHost(domainId, **attrs, **params)	PUT /v1.0/cdn/domains/{domain_id}/originhost
	getOriginHost(domainId, **params)	GET /v1.0/cdn/domains/{domain_id}/originhost
	setOriginRange(domainId, **attrs)	PUT /v1.0/cdn/domains/{domainId}/range-switch
	setFollow302(domainId, **attrs)	PUT /v1.0/cdn/domains/{domainId}/follow302-switch
	setReferer(domainId, **attrs, **params)	PUT /v1.0/cdn/domains/{domain_id}/referer
	getReferer(domainId, **params)	GET /v1.0/cdn/domains/{domain_id}/referer
	getIpAcl(domainId)	GET /v1.0/cdn/domains/{domainId}/ip-acl
	setIpAcl(domainId, **attrs)	PUT /v1.0/cdn/domains/{domainId}/ip-acl

Interface	Method	API
	setCacheConfig(domainId, **attrs, **params)	PUT /v1.0/cdn/domains/{domain_id}/cache
	getCacheConfig(domainId, **params)	GET /v1.0/cdn/domains/{domain_id}/cache
	setHttpsInfo(domainId, **attrs, **params)	PUT /v1.0/cdn/domains/{domain_id}/https-info
	getHttpsInfo(domainId, **params)	GET /v1.0/cdn/domains/{domain_id}/https-info
	queryCdnIPs(**attrs)	GET /v1.0/cdn/ip-info
	setResponseHeader(domainId, **attrs)	PUT /v1.0/cdn/domains/{domainId}/response-header
	getResponseHeader(domainId)	GET /v1.0/cdn/domains/{domainId}/response-header
	createRefreshTask(**attrs, **params)	POST /v1.0/cdn/refreshtasks
	createPreheatingTask(**attrs, **params)	POST /v1.0/cdn/preheatingtasks
	queryTasks(**params)	GET /v1.0/cdn/historytasks
	getTaskDetail(taskId, **params)	GET /v1.0/cdn/historytasks/{task_id}/detail
StatisticService	queryTopUrl(**query)	GET /v1.0/cdn/statistics/top-url
	queryDomainItemDetails(**query)	GET /v1.0/cdn/statistics/domain-item-details
	queryDomainItemLocationDetails(**query)	GET /v1.0/cdn/statistics/domain-item-location-details
	queryTotalNetworkTraffic(**query)	GET /v1.0/cdn/statistics/flux
	queryDetailsOfNetworkTraffic(**query)	GET /v1.0/cdn/statistics/flux-detail
	queryPeakBandwidth(**query)	GET /v1.0/cdn/statistics/bandwidth
	queryDetailsOfNetworkBandwidth(**query)	GET /v1.0/cdn/statistics/bandwidth-detail
	queryConsumptionSummary(**query)	GET /v1.0/cdn/statistics/domain-summary
	queryConsumptionSummaryDetails(**query)	GET /v1.0/cdn/statistics/domain-summary-detail

Interface	Method	API
	queryDomainConsumptions(**query)	GET /v1.0/cdn/statistics/domain
	queryRegionDetailSummry(**query)	GET /v1.0/cdn/statistics/region-detail-summary
	queryCarrierDetailSummry(**query)	GET /v1.0/cdn/statistics/carrier-detail-summary
	queryRegionCarrierDomain(**query)	GET /v1.0/cdn/statistics/region-carrier-domain
	queryRegionCarrierDetail(**query)	GET /v1.0/cdn/statistics/region-carrier-detail
LogService	queryLogs(domainName, queryDate, pageSize, pageNumber, enterpriseProjectId)	GET /v1.0/cdn/logs

6.1.17 FGS

The SDK interfaces based on the FGS v1.0 JAVA API are as follows. For details about the invoking methods, see the sample codes.

Resource	Method	API
Functions	FunctionMetadata.Functions listFunction(int marker, int maxItems)	GET /v1.0/{project_id}/fss/functions?marker={marker}&maxitems={maxitems}
	FunctionMetadata getFunctionMetadata(String function_urn)	GET /v1.0/{project_id}/fss/functions/{function_urn}/config
	FunctionMetadata getFunctionCode(String function_urn)	GET /v1.0/{project_id}/fss/functions/{function_urn}/code
	FunctionMetadata createFunction(FunctionMetadata functionMetadata)	POST /v1.0/{project_id}/fss/functions
	ActionResponse deleteFunction(String function_urn)	DELETE /v1.0/{project_id}/fss/functions/{function_urn}

Resource	Method	API
	FunctionMetadata updateFunctionCode(String function_urn, FunctionMetadata functionMetadata)	PUT /v1.0/{project_id}/fss/functions/{function_urn}/code
	FunctionMetadata updateFunctionConfig(String function_urn, FunctionMetadata functionMetadata)	PUT /v1.0/{project_id}/fss/functions/{function_urn}/config
	String invokeFunction(String function_urn, Map<?, ?> data)	POST /v1.0/{project_id}/fss/functions/{function_urn}/invocations
	String asyncInvokeFunction(String function_urn, Map<?, ?> data)	POST /v1.0/{project_id}/fss/functions/{function_urn}/invocations-async
Versions	FunctionMetadata PublishVersion(String function_urn, FunctionMetadata fmd);	POST /v1.0/{project_id}/fss/functions/{function_urn}/versions
	FunctionMetadata.FunctionVersions listFunctionVersions(String function_urn, int marker, int maxItems)	GET /v1.0/{project_id}/fss/functions/{function_urn}/versions?marker={marker}&maxitems={maxitems}
	FunctionVersionAlias createVersionAlias(String function_urn, FunctionVersionAlias functionVersionAlias)	POST /v1.0/{project_id}/fss/functions/{function_urn}/aliases
	FunctionVersionAlias updateVersionAlias (String function_urn, FunctionVersionAlias functionVersionAlias)	PUT /v1.0/{project_id}/fss/functions/{function_urn}/aliases/{alias_name}
	void DeleteVersionAlias(String function_urn, String alias_name);	DELETE /v1.0/{project_id}/fss/functions/{function_urn}/aliases/{alias_name}
	FunctionVersionAlias GetVersionAlias(String function_urn, String alias_name);	GET /v1.0/{project_id}/fss/functions/{function_urn}/aliases/{alias_name}

Resource	Method	API
	List<FunctionVersionAlias> ListVersionAlias(String function_urn)	GET /v1.0/{project_id}/fss/functions/{function_urn}/aliases
Triggers	FunctionTrigger[] listTriggersForFunction(String function_urn)	GET /v1.0/{project_id}/fss/triggers/{function_urn}
	FunctionTriggers CreateTriggerInstance(String function_urn, FunctionTriggers functionTriggers)	POST /v1.0/{project_id}/fss/triggers/{function_urn}
	void DeleteTrigger(String function_urn, String trigger_type_code, String trigger_id)	DELETE /v1.0/{project_id}/fss/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
	FunctionTriggers<?> GetTriggerInstance(String function_urn, String trigger_type_code, String trigger_id)	GET /v1.0/{project_id}/fss/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
	void DeleteAllTriggersFor-Function(String function_urn);	DELETE /v1.0/{project_id}/fss/triggers/{function_urn}

The SDK interfaces based on the FGS v2.0 JAVA API are as follows. For details about the invoking methods, see the sample codes.

Resource	Method	API
Functions	FunctionMetadata.Functions listFunction(int marker, int maxItems)	GET /v2/{project_id}/fgs/functions?marker={marker}&maxitems={maxitems}
	FunctionMetadata.Functions listFunction()	GET /v2/{project_id}/fgs/functions
	FunctionMetadata getFunctionMetadata(String function_urn)	GET /v2/{project_id}/fgs/functions/{function_urn}/config
	FunctionMetadata getFunctionCode(String function_urn)	GET /v2/{project_id}/fgs/functions/{function_urn}/code

Resource	Method	API
	FunctionMetadata createFunction(FunctionMetadata functionMetadata)	POST /v2/{project_id}/fgs/functions
	ActionResponse deleteFunction(String function_urn)	DELETE /v2/{project_id}/fgs/functions/{function_urn}
	FunctionMetadata updateFunctionCode(String function_urn, FunctionMetadata functionMetadata)	PUT /v2/{project_id}/fgs/functions/{function_urn}/code
	FunctionMetadata updateFunctionConfig(String function_urn, FunctionMetadata functionMetadata)	PUT /v2/{project_id}/fgs/functions/{function_urn}/config
	FunctionMetadata createFunctionVersion(String function_urn, FunctionMetadata fmd);	POST /v2/{project_id}/fgs/functions/{function_urn}/versions
	FunctionMetadata.FunctionVersions listFunctionVersion(String function_urn, int marker, int maxItems)	GET /v2/{project_id}/fgs/functions/{function_urn}/versions?marker={marker}&maxitems={maxitems}
	FunctionMetadata.FunctionVersions listFunctionVersion(String function_urn)	GET /v2/{project_id}/fgs/functions/{function_urn}/versions
	FuncInvocations invokeFunction(String function_urn, Map<?, ?> data)	POST /v2/{project_id}/fgs/functions/{function_urn}/invocations
	FuncInvocations asyncInvokeFunction(String function_urn, Map<?, ?> data)	POST /v2/{project_id}/fgs/functions/{function_urn}/invocations-async
Versions	FunctionVersionAlias createVersionAlias(String function_urn, FunctionVersionAlias functionVersionAlias)	POST /v2/{project_id}/fgs/functions/{function_urn}/aliases

Resource	Method	API
	FunctionVersionAlias updateVersionAlias (String function_urn, FunctionVersionAlias functionVersionAlias)	PUT /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
	ActionResponse deleteVersionAlias(String function_urn, String alias_name);	DELETE /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
	FunctionVersionAlias getVersionAlias(String function_urn, String alias_name);	GET /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
	List<FunctionVersionAlias> listVersionAlias(String function_urn)	GET /v2/{project_id}/fgs/functions/{function_urn}/aliases
Triggers	FunctionTrigger[] listTriggersForFunction(String function_urn)	GET /v2/{project_id}/fgs/triggers/{function_urn}
	FunctionTriggers createTriggerInstance(String function_urn, FunctionTriggers functionTriggers)	POST /v2/{project_id}/fgs/triggers/{function_urn}
	ActionResponse deleteTrigger(String function_urn, String trigger_type_code, String trigger_id)	DELETE /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
	FunctionTriggers<?> getTriggerInstance(String function_urn, String trigger_type_code, String trigger_id)	GET /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
	ActionResponse deleteAllTriggersForFunction(String function_urn);	DELETE /v2/{project_id}/fgs/triggers/{function_urn}

6.1.18 TMS

The SDK interfaces based on TMS v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
TagService	String create(List<PredefineTagRequest> tagList)	POST /v1.0/predefine_tags/action Link
	String dekete(List<PredefineTagRequest> tagList)	POST/v1.0/predefine_tags/action Link
	PredefineTags list()	GET/v1.0/predefine_tags Link
	PredefineTags list(TagFilterOption option)	GET/v1.0/predefine_tags Link
	ActionResponse modify(PredefineTagRequest oldTag, PredefineTagRequest newTag)	PUT/v1.0/predefine_tags Link

6.1.19 EPS

The SDK interfaces based on EPS v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
EPService	EPCreateResponse create(EPCreateRequest createRequest)	POST/v1.0/enterprise-projects Link
	EPListResponse list()	GET/v1.0/enterprise-projects Link
	EPListResponse list(Map<String, String> queryParams)	GET/v1.0/enterprise-projects Link
	EPQueryResponse get(String epID)	GET/v1.0/enterprise-projects/{id} Link
	EPQueryResponse modify(String epID, EPModifyRequest modifyRequest)	PUT/v1.0/enterprise-projects/{id} Link
	EPQuotaResponse quotas()	GET/v1.0//enterprise-projects/quotas Link

Interface	Method	API
	ActionResponse action(String epID, EPActionRequest actionRequest)	POST/v1.0/enterprise-projects/{id}/action **Link**
	EPResourceFilterResponse filterResource(String epID, EPResourceFilterRequest filterRequest)	POST/v1.0/enterprise-projects/{id}/resources/filter **Link**
	ActionResponse migrateResource(String epID, EPResourceActionRequest actionRequest)	POST/v1.0/enterprise-projects/{id} /resources-migrate **Link**

6.1.20 RDS

The SDK interfaces based on the RDS v3 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
InstanceManageService	CreateInstanceResponse create(CreateInstanceRequest createRequest)	POST /v3/{project_id}/instances **Link**
	ListInstanceResponse list()	GET /v3/{project_id}/instances **Link**
	ListInstanceResponse list(Map<String, String> queryParams)	GET /v3/{project_id}/instances **Link**
	ActionResponse delete(String instanceId)	DELETE /v3/{project_id}/instances/{instanceId} **Link**
	InstanceCommonResponse restart(RestartInstanceRequest request, String instanceId)	PUT /v3/{project_id}/instances/{instanceId}/action **Link**
	InstanceCommonResponse singleToHa(SingleToHaRdsRequest request, String instanceId)	GET /v3/{project_id}/instances/{instanceId}/action **Link**

Interface	Method	API
	InstanceCommonResponse resizeFlavor(ResizeFlavorRequest request, String instanceId)	POST /v3/{project_id}/instances/{instanceId}/action **Link**
	InstanceCommonResponse enlargeVolume(EnlargeVolumeRequest request, String instanceId)	POST /v3/{project_id}/instances/{instanceId}/action **Link**
	ErrorLogResponse listErrorLog(Map<String, String> queryParams, String instanceId)	GET /v3/{project_id}/instances/{instanceId} /errorlog **Link**
	SlowLogListResponse listSlowLog(Map<String, String> queryParams, String instanceId)	GET /v3/{project_id}/instances/{instanceId}/slowlog **Link**
	FlavorList listFlavors(Map<String, String> filteringParams, String engineName)	GET /v3/{project_id}/flavors/{engineName} **Link**
BackupsAndRestoreService	ManualBackupResponse create(ManualBackupRequest request)	POST /v3/{project_id}/backups **Link**
	BackupsResponse list(String instanceId, Map<String, String> filterParams)	GET /v3/{project_id}/backups **Link**
	ActionResponse delete(String backupId)	DELETE /v3/{project_id}/backups/{backupId} **Link**

6.2 Python

6.2.1 IAM

The SDK interfaces based on the IAM API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
AuthToken Operations	create_authtoken(self, attr, nocatalog=None)	POST /v3/auth/tokens • Obtaining a User Token Through Password Authentication **Link** • Obtaining a User Token Through Password and Virtual MFA Authentication **Link** • Obtaining an Agency Token **Link**
	validate_authtoken(self, x_subject_token, nocatalog=None):	GET /v3/auth/tokens **Link**
Securitytoken Operations	create_securitytoken(self, **attrs)	POST /v3.0/OS-CREDENTIAL/securitytokens • Obtaining a Temporary Access Key and Security Token Through an Agency **Link** • Obtaining a Temporary Access Key and Security Token Through a Token **Link**
Credential Operations	create_credential(self, **attrs)	POST /v3.0/OS-CREDENTIAL/credentials **Link**
	credentials(self, **query)	GET /v3.0/OS-CREDENTIAL/credentials **Link**
	get_credential(self, access_key)	GET /v3.0/OS-CREDENTIAL/credentials/{access_key} **Link**
	update_credential(self, access_key, **attrs)	PUT /v3.0/OS-CREDENTIAL/credentials/{access_key} **Link**
	delete_credential(self, access_key)	DELETE /v3.0/OS-CREDENTIAL/credentials/{access_key} **Link**
Region Operations	regions(self, **query)	GET /v3/regions **Link**
	get_region(self, region)	GET /v3/regions/{region_id} **Link**

Interface	Method	API
Project Operations	projects(self, **query)	GET /v3/projects **Link**
	list_user_projects(self, user_id)	GET /v3/users/{user_id}/projects **Link**
	get_project_scopes(self)	GET /v3/auth/projects **Link**
	create_project(self, **attrs)	POST /v3/projects **Link**
	update_project(self, project, **attrs)	PATCH /v3/projects/{project_id} **Link**
	get_project(self, project)	GET /v3/projects/{project_id} **Link**
Project Operations (iam)	update_project_status(self, project_id, attrs)	PUT /v3-ext/projects/{project_id} **Link**
	get_project_details_and_status(self, project_id)	GET /v3-ext/projects/{project_id} **Link**
Domain Operations	get_domain_scopes(self)	GET /v3/auth/domains **Link**
PasswordConfig Operations	get_password_config(self, domain_id)	GET /v3/domains/{domain_id}/config/security_compliance **Link**
	get_password_config_by_option(self, domain_id, option)	GET /v3/domains/{domain_id}/config/security_compliance/{option} **Link**
User Operations (identity)	users(self, **query)	GET /v3/users **Link**
	get_user(self, user)	GET /v3/users/{user_id} **Link**
	list_group_users(self, group_id)	GET /v3/groups/{group_id}/users **Link**
	create_user(self, **attrs)	POST /v3/users **Link**

Interface	Method	API
	change_password(self, user_id, **attrs)	POST /v3/users/{user_id}/password Link
	update_user(self, user, **attrs)	PATCH /v3/users/{user_id} Link
	delete_user(self, user, ignore_missing=True)	DELETE /v3/users/{user_id} Link
	remove_user_from_group(self, group_id, user_id)	DELETE /v3/groups/{group_id}/users/{user_id} Link
User Operations (iam)	query_user_details(self, user_id)	GET /v3.0/OS-USER/users/{user_id} Link
	create_user(self, **user)	POST /v3.0/OS-USER/users Link
	update_user_information(self, user_id, **user)	PUT /v3.0/OS-USER/users/{user_id}/info Link
	update_user_information_by_admin(self, user_id, **user)	PUT /v3.0/OS-USER/users/{user_id} Link
Group Operations	groups(self, **query)	GET /v3/groups Link
	get_group(self, group)	GET /v3/groups/{group_id} Link
	create_group(self, **attrs)	POST /v3/groups Link
	update_group(self, group, **attrs)	PATCH /v3/groups/{group_id} Link
	delete_group(self, group, ignore_missing=True)	DELETE /v3/groups/{group_id} Link
	check_group_user(self, group_id, user_id)	HEAD /v3/groups/{group_id}/users/{user_id} Link

Interface	Method	API
	add_user_to_group(self, group_id, user_id)	PUT /v3/groups/{group_id}/users/{user_id} **Link**
	list_user_groups(self, user_id)	GET /v3/users/{user_id}/groups **Link**
Role Operations	roles(self, **query)	GET /v3/roles **Link**
	get_role(self, role)	GET /v3/roles/{role_id} **Link**
	list_domain_user_group_role(self, domain_id, group_id)	GET /v3/domains/{domain_id}/groups/{group_id}/roles **Link**
	list_project_user_group_role(self, project_id, group_id)	GET /v3/projects/{project_id}/groups/{group_id}/roles **Link**
	grant_domain_group_role(self, domain_id, group_id, role_id)	PUT /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id} **Link**
	grant_project_group_role(self, project_id, group_id, role_id)	PUT /v3/projects/{project_id}/groups/{group_id}/roles/{role_id} **Link**
	check_domain_group_role(self, domain_id, group_id, role_id)	HEAD /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id} **Link**
	check_project_group_role(self, project_id, group_id, role_id)	HEAD /v3/projects/{project_id}/groups/{group_id}/roles/{role_id} **Link**
	delete_domain_group_role(self, domain_id, group_id, role_id)	DELETE /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id} **Link**
	delete_project_group_role(self, project_id, group_id, role_id)	DELETE /v3/projects/{project_id}/groups/{group_id}/roles/{role_id} **Link**

Interface	Method	API
	grant_all_projects_group_role(self, domain_id, group_id, role_id)	PUT /v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects **Link**
Customrole Operations	custom_roles(self)	GET /v3.0/OS-ROLE/roles **Link**
	get_custom_role(self, role_id)	GET /v3.0/OS-ROLE/roles/{role_id} **Link**
	create_custom_role(self, **attrs)	POST /v3.0/OS-ROLE/roles • Creating a Custom Cloud Service Policy **Link** • Creating a Custom Agency Policy **Link**
	update_custom_role(self, role_id, **attrs)	PATCH /v3.0/OS-ROLE/roles/{role_id} • Modifying a Custom Cloud Service Policy **Link** • Modifying a Custom Agency Policy **Link**
	delete_custom_role(self, role_id, ignore_missing=True)	DELETE /v3.0/OS-ROLE/roles/{role_id} **Link**
Agency Operations	agencies(self, **query)	GET /v3.0/OS-AGENCY/agencies **Link**
	get_agency(self, agency_id)	GET /v3.0/OS-AGENCY/agencies/{agency_id} **Link**
	create_agency(self, **attrs)	POST /v3.0/OS-AGENCY/agencies **Link**
	update_agency(self, agency_id, **attrs)	PUT /v3.0/OS-AGENCY/agencies/{agency_id} **Link**
	delete_agency(self, agency_id)	DELETE /v3.0/OS-AGENCY/agencies/{agency_id} **Link**

Interface	Method	API
	list_domain_agency_role(self, domain_id, agency_id)	GET /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles **Link**
	list_project_agency_role(self, project_id, agency_id)	GET /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles **Link**
	grant_domain_agency_role(self, domain_id, agency_id, role_id)	PUT /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id} **Link**
	grant_project_agency_role(self, project_id, agency_id, role_id)	PUT /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id} **Link**
	check_domain_agency_role(self, domain_id, agency_id, role_id)	HEAD /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id} **Link**
	check_project_agency_role(self, project_id, agency_id, role_id)	HEAD /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id} **Link**
	delete_domain_agency_role(self, domain_id, agency_id, role_id)	DELETE /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id} **Link**
	delete_project_agency_role(self, project_id, agency_id, role_id)	DELETE /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id} **Link**
Version Operations	get_version_of_keystone(self)	GET / **Link**
	get_version3_of_keystone(self)	GET /v3 **Link**
Service Operations	services(self, **query)	GET /v3/services **Link**

Interface	Method	API
	get_service(self, service)	GET /v3/services/{service_id} Link
	get_service_catalog(self)	GET /v3/auth/catalog Link
Endpoint Operations	endpoints(self, **query)	GET /v3/endpoints Link
	get_endpoint(self, endpoint)	GET /v3/endpoints/{endpoint_id} Link

6.2.2 IMS

The SDK interfaces based on the Glance v2 API are as follows. Invocation example: conn.image.upload_image()

Interface	Method	API
Image Operations	upload_image(self, container_format=None, disk_format=None, data=None, **attrs)	POST /v2/images PUT /v2/images/{image_id}/file
	delete_image(self, image, ignore_missing=True)	DELETE /v2/images/{image_id}
	find_image(self, name_or_id, ignore_missing=True)	GET /v2/images
	get_image(self, image)	GET /v2/images/{image_id}
	images(self, **query)	GET /v2/images
	add_tag(self, image, tag)	PUT /v2/images/{image_id}/tags/{tag}
	remove_tag(self, image, tag)	DELETE /v2/images/{image_id}/tags/{tag}
Member Operations	add_member(self, image, **attrs)	POST /v2/images/{image_id}/members
	remove_member(self, member, image, ignore_missing=True)	DELETE /v2/images/{image_id}/members/{member_id}
	find_member(self, name_or_id, image, ignore_missing=True)	GET /v2/images/{image_id}/members

Interface	Method	API
	get_member(self, member, image)	GET /v2/images/{image_id}/members/{member_id}
	members(self, image)	GET /v2/images/{image_id}/members
	update_member(self, member, image, **attrs)	PUT /v2/images/{image_id}/members/{member_id}

The SDK interfaces based on the IMS v2 API are as follows. Invocation example: conn.ims.create_cloudimage()

Interface	Method	API
Cloudimage Operations	cloudimages(self, **query)	GET /v2/cloudimages
	update_cloudimage(self, cloudimage_id, **data)	PATCH /v2/cloudimages/{image_id}
	create_cloudimage(self, **data)	POST /v2/cloudimages/action
	get_job(self, job_id):	GET /v1/{project_id}/jobs/{job_id}

6.2.3 VPC

The SDK interfaces based on the VPC v1 API are as follows. Invocation example: conn.vpc.create_network().

Interface	Method	API
vpc	vpcs(self, **query)	GET /v1/{project_id}/vpcs **Link**
	get_vpc(self, vpc)	GET /v1/{project_id}/vpcs/{vpc_id} **Link**
	create_vpc(self, **attrs)	POST /v1/{project_id}/vpcs **Link**
	update_vpc(self, vpc, **attrs)	PUT /v1/{project_id}/vpcs/{vpc_id} **Link**

Interface	Method	API
	delete_vpc(self, vpc, ignore_missing=True)	DELETE /v1/{project_id}/vpcs/{vpc_id} **Link**
	find_vpc(self, name_or_id, ignore_missing=True)	• GET /v1/{project_id}/vpcs/{vpc_id} **Link** • GET /v1/{project_id}/vpcs **Link**
subnet	subnets(self, **query)	GET /v1/{project_id}/subnets **Link**
	get_subnet(self, subnet)	GET /v1/{project_id}/subnets/{subnet_id} **Link**
	create_subnet(self, **attrs)	POST /v1/{project_id}/subnets **Link**
	update_subnet(self, subnet, vpc_id, **attrs)	PUT /v1/{project_id}/vpcs/{vpc_id}/subnets/{subnet_id} **Link**
	delete_subnet(self, subnet, vpc_id, ignore_missing=True)	DELETE /v1/{project_id}/vpcs/{vpc_id}/subnets/{subnet_id} **Link**
	find_subnet(self, name_or_id, ignore_missing=True)	• GET /v1/{project_id}/subnets/{subnet_id} **Link** • GET /v1/{project_id}/subnets **Link**
public_ip	public_ips(self, **query)	GET /v1/{project_id}/publicips **Link**
	get_public_ip(self, public_ip)	GET /v1/{project_id}/publicips/{publicip_id} **Link**
	create_public_ip(self, **attrs)	POST /v1/{project_id}/publicips **Link**
	update_public_ip(self, public_ip, **attrs)	PUT /v1/{project_id}/publicips/{publicip_id} **Link**

Interface	Method	API
	delete_public_ip(self, public_ip, ignore_missing=True)	DELETE /v1/{project_id}/publicips/{publicip_id} **Link**
	find_public_ip(self, name_or_id, ignore_missing=True)	• GET /v1/{project_id}/publicips/{publicip_id} **Link** • GET /v1/{project_id}/publicips **Link**
private_ip	private_ips(self, subnet, **query)	GET /v1/{project_id}/subnets/{subnet_id}/privateips **Link**
	get_private_ip(self, private_ip)	GET /v1/{project_id}/privateips/{privateip_id} **Link**
	create_private_ip(self, **attrs)	POST /v1/{project_id}/privateips **Link**
	create_private_ips(self, *private_ips)	POST /v1/{project_id}/privateips **Link**
	delete_private_ip(self, private_ip, ignore_missing=True)	DELETE /v1/{project_id}/privateips/{privateip_id} **Link**
	find_private_ip(self, name_or_id, subnet_id, ignore_missing=True)	• GET /v1/{project_id}/privateips/{privateip_id} **Link** • GET /v1/{project_id}/subnets/{subnet_id}/privateips **Link**
port	ports(self, **query)	GET /v1/{project_id}/ports **Link**
	get_port(self, port)	GET /v1/{project_id}/ports/{port_id} **Link**
	create_port(self, **attrs)	POST /v1/{project_id}/ports **Link**
	update_port(self, port, **attrs)	PUT /v1/{project_id}/ports/{port_id} **Link**

Interface	Method	API
	delete_port(self, port, ignore_missing=True)	DELETE /v1/{project_id}/ports/{port_id} **Link**
	find_port(self, name_or_id, ignore_missing=True)	• GET /v1/{project_id}/ports/{port_id} **Link** • GET /v1/{project_id}/ports **Link**
bandwidth	bandwidths(self, **query)	GET /v1/{project_id}/bandwidths **Link**
	get_bandwidth(self, bandwidth)	GET /v1/{project_id}/bandwidths/{bandwidth_id} **Link**
	update_bandwidth(self, bandwidth, **attrs)	PUT /v1/{project_id}/bandwidths/{bandwidth_id} **Link**
	find_bandwidth(self, name_or_id, ignore_missing=True)	• GET /v1/{project_id}/bandwidths/{bandwidth_id **Link** • GET /v1/{project_id}/bandwidths **Link**
security_group	security_groups(self, **query)	GET /v1/{project_id}/security-groups **Link**
	create_security_group(self, **attrs)	POST /v1/{project_id}/security-groups **Link**
	delete_security_group(self, security_group, ignore_missing=True)	DELETE /v1/{project_id}/security-groups/{security_group_id} **Link**
	get_security_group(self, security_group)	GET /v1/{project_id}/security-groups/{security_group_id} **Link**

Interface	Method	API
	find_security_group(self, name_or_id, ignore_missing=True)	• GET /v1/{project_id}/security-groups/{security_group_id} **Link** • GET /v1/{project_id}/security-groups **Link**
security_group_rule	create_security_group_rule(self, **attrs)	POST /v1/{project_id}/security-group-rules **Link**
	delete_security_group_rule(self, security_group_rule, ignore_missing=True)	DELETE /v1/{project_id}/security-group-rules/{rules_security_groups_id} **Link**
	get_security_group_rule(self, security_group_rule)	GET /v1/{project_id}/security-group-rules/{rules_security_groups_id} **Link**
	security_group_rules(self, **query)	GET /v1/{project_id}/security-group-rules **Link**
	find_security_group_rule(self, name_or_id, ignore_missing=True)	• GET /v1/{project_id}/security-group-rules/{rules_security_groups_id} **Link** • GET /v1/{project_id}/security-group-rules **Link**
quota	quotas(self, **query)	GET /v1/{project_id}/quotas **Link**

The SDK interfaces based on the VPC v2.0 API are as follows.

Invocation example: conn.vpc.create_publicip_ext()

Interface	Method	API
sharebandwidth	create_sharebandwidth(self, **data)	POST /v2.0/{project_id}/bandwidths **Link**

Interface	Method	API
	delete_sharebandwidth(self, bandwidth_id, ignore_missing=True)	DELETE /v2.0/{project_id}/bandwidths/{bandwidth_id} **Link**
	create_batch_sharebandwidth(self, **data)	POST /v2.0/{project_id}/batch-bandwidths **Link**
	insert_ip_to_bandwidth(self, bandwidth_id, **data)	POST /v2.0/{project_id}/bandwidths/{bandwidth_id}/insert **Link**
	remove_ip_from_bandwidth(self, bandwidth_id, **data)	POST /v2.0/{project_id}/bandwidths/{bandwidth_id}/remove **Link**
Eip Operations	create_publicip_ext(self, **attrs)	POST /v2.0/{project_id}/publicips **Link**
Bandwidth Operations	update_bandwidth_ext(self, bandwidth_id, **attrs)	PUT /v2.0/{project_id}/bandwidths/{bandwidth_id} **Link**

The SDK interfaces based on the Neutron v2.0 API are as follows.

Invocation example: conn.network.create_network ()

Interface	Method	API
Floating IP Operations	create_ip(self, **attrs)	POST /v2.0/floatingips **Link**
	delete_ip(self, floating_ip, ignore_missing=True)	DELETE /v2.0/floatingips/{floatingip_id} **Link**
	find_available_ip(self)	GET /v2.0/floatingips **Link**
	find_ip(self, name_or_id, ignore_missing=True)	GET /v2.0/floatingips **Link**

Interface	Method	API
	get_ip(self, floating_ip)	GET /v2.0/floatingips/{floatingip_id} **Link**
	ips(self, **query)	GET /v2.0/floatingips **Link**
	update_ip(self, floating_ip, **attrs)	PUT /v2.0/floatingips/{floatingip_id} **Link**
Network Operations	create_network(self, **attrs)	POST /v2.0/networks **Link**
	delete_network(self, network, ignore_missing=True)	DELETE /v2.0/networks/{network_id} **Link**
	find_network(self, name_or_id, ignore_missing=True)	GET /v2.0/networks **Link**
	get_network(self, network)	GET /v2.0/networks/{network_id} **Link**
	networks(self, **query)	GET /v2.0/networks **Link**
	update_network(self, network, **attrs)	PUT /v2.0/networks/{network_id} **Link**
Port Operations	create_port(self, **attrs)	POST /v2.0/ports **Link**
	delete_port(self, port, ignore_missing=True)	DELETE /v2.0/ports/{port_id} **Link**
	find_port(self, name_or_id, ignore_missing=True)	GET /v2.0/ports **Link**
	get_port(self, port)	GET /v2.0/ports/{port_id} **Link**
	ports(self, **query)	GET /v2.0/ports **Link**

Interface	Method	API
	update_port(self, port, **attrs)	PUT /v2.0/ports/{port_id} Link
Router Operations	create_router(self, **attrs)	POST /v2.0/router Link
	delete_router(self, router, ignore_missing=True)	DELETE /v2.0/routers/{router_id} Link
	find_router(self, name_or_id, ignore_missing=True)	GET /v2.0/routers Link
	get_router(self, router)	GET /v2.0/routers/{router_id} Link
	routers(self, **query)	GET /v2.0/routers Link
	update_router(self, router, **attrs)	PUT /v2.0/routers/{router_id} Link
	add_interface_to_router(self, router, subnet_id=None, port_id=None)	PUT /v2.0/routers/{router_id}/add_router_interface Link
	remove_interface_from_router(self, router, subnet_id=None, port_id=None)	PUT /v2.0/routers/{router_id}/remove_router_interface Link
Security Group Operations	create_security_group(self, **attrs)	POST /v2.0/security-groups Link
	delete_security_group(self, security_group, ignore_missing=True)	DELETE /v2.0/security-groups/{security_group_id} Link
	find_security_group(self, name_or_id, ignore_missing=True)	GET /v2.0/security-groups Link

Interface	Method	API
	get_security_group(self, security_group)	GET /v2.0/security-groups/{security_group_id} **Link**
	security_groups(self, **query)	GET /v2.0/security-groups **Link**
	update_security_group(self, security_group, **attrs)	PUT /v2.0/security-groups/{security_group_id} **Link**
	security_group_open_port(self, sgid, port, protocol='tcp')	POST /v2.0/security-group-rules **Link**
	security_group_allow_ping(self, sgid)	POST /v2.0/security-group-rules **Link**
	create_security_group_rule(self, **attrs)	POST /v2.0/security-group-rules **Link**
	delete_security_group_rule(self, security_group_rule, ignore_missing=True)	DELETE /v2.0/security-group-rules/{security_group_rule_id} **Link**
	find_security_group_rule(self, name_or_id, ignore_missing=True)	GET /v2.0/security-group-rules **Link**
	get_security_group_rule(self, security_group_rule)	GET /v2.0/security-group-rules/{security_group_rule_id} **Link**
	security_group_rules(self, **query)	GET /v2.0/security-group-rules **Link**
Subnet Operations	create_subnet(self, **attrs)	POST /v2.0/subnets **Link**
	delete_subnet(self, subnet, ignore_missing=True)	DELETE /v2.0/subnets/{subnet_id} **Link**

Interface	Method	API
	find_subnet(self, name_or_id, ignore_missing=True)	GET /v2.0/subnets **Link**
	get_subnet(self, subnet)	GET /v2.0/subnets/{subnet_id} **Link**
	subnets(self, **query)	GET /v2.0/subnets **Link**
	get_subnet_ports(self, subnet_id)	GET /v2.0/ports **Link**
	update_subnet(self, subnet, **attrs)	PUT /v2.0/subnets/{subnet_id} **Link**

6.2.4 ECS

The SDK interfaces based on the Nova v2 API are as follows. Invocation example: conn.compute.create_server()

Interface	Method	API
Flavor Operations	find_flavor(self, name_or_id, ignore_missing=True)	• name: GET /v2/{project_id}/flavors **Link** • id: GET /v2/{project_id}/flavors/{flavor_id} **Link**
	get_flavor(self, flavor)	GET /v2/{project_id}/flavors/{flavor_id} **Link**
	query_flavor_extra_specs(self, flavor_id)	GET /v2/{project_id}/flavors/{flavors_id}/os-extra_specs **Link**
	flavors(self, details=True, paginated=True, **query)	GET /v2/{project_id}/flavors/detail **Link**
Image Operations	delete_image(self, image, ignore_missing=True)	DELETE /v2/{project_id}/images/{image_id} **Link**

Interface	Method	API
	find_image(self, name_or_id, ignore_missing=True)	name: GET /v2/{project_id}/images **Link** Image_id: GET /v2/{project_id}/images/{image_id} **Link**
	get_image(self, image)	GET /v2/{project_id}/images/{image_id} **Link**
	images(self, details=True, **query)	GET /v2/{project_id}/images/detail **Link**
	get_image_metadata(self, image)	GET /v2/{project_id}/images/{image_id}/metadata **Link**
Keypair Operations	create_keypair(self, **attrs)	POST /v2/{project_id}/os-keypairs **Link**
	delete_keypair(self, keypair, ignore_missing=True)	DELETE /v2/{project_id}/os-keypairs/{keypair_name} **Link**
	get_keypair(self, keypair)	GET /v2/{project_id}/os-keypairs/{keypair_name} **Link**
	find_keypair(self, name, ignore_missing=True)	Keypair_name: GET /v2/{project_id}/os-keypairs/{keypair_name} **Link** Name: GET /v2/{project_id}/os-keypairs **Link**
	keypairs(self)	GET /v2/{project_id}/os-keypairs **Link**
Server Operations	create_server(self, **attrs)	POST /v2/{project_id}/servers **Link**

Interface	Method	API
	delete_server(self, server, ignore_missing=True, force=False)	DELETE /v2/{project_id}/servers/{server_id} **Link**
	find_server(self, name_or_id, ignore_missing=True)	Name: GET /v2/{project_id}/servers **Link** Server_id: GET /v2/{project_id}/servers/{server_id} **Link**
	get_server(self, server)	GET /v2/{project_id}/servers/{server_id} **Link**
	servers_list(self, details=True, paginated=True, headers=None, **query)	• details=True: GET /v2/{project_id}/servers/detail **Link** • details=False: GET /v2/{project_id}/servers **Link**
	create_server_image(self, server, name, metadata=None)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	add_floating_ip_to_server(self, server, address, fixed_address=None)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	remove_floating_ip_from_server(self, server, address)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	lock_server(self, server)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	unlock_server(self, server)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	start_server(self, server)	POST /v2/{project_id}/servers/{server_id}/action **Link**

Interface	Method	API
	stop_server(self, server)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	wait_for_server(self, server, status='ACTIVE', failures=['ERROR'], interval=2, wait=120)	GET /v2/{project_id}/servers/{server_id} **Link**
	instance_actions(self, server_id)	GET /v2/{project_id}/servers/{server_id}/os-instance-actions **Link**
	get_instance_action(self, server_id, request_id)	GET /v2/{project_id}/servers/{server_id}/os-instance-actions/{request_id} **Link**
	get_server_console_output(self, server_id, lines)	POST /v2/{project_id}/servers/{server_id}/action **Link**
Server Interface Operations	create_server_interface(self, server, **attrs)	POST /v2/{project_id}/servers/{server_id}/os-interface **Link**
	delete_server_interface(self, server_interface, server=None, ignore_missing=True)	DELETE /v2/{project_id}/servers/{server_id}/os-interface/{port_id} **Link**
	get_server_interface(self, server_interface, server=None)	GET /v2/{project_id}/servers/{server_id}/os-interface/{port_id} **Link**
	server_interfaces(self, server)	GET /v2/{project_id}/servers/{server_id}/os-interface **Link**
Server IPs Operations	server_ips(self, server, network_label=None)	GET /v2/{project_id}/servers/{server_id}/ips **Link**
Availability Zone Operations	availability_zones(self, details=False)	GET /v2/{project_id}/os-availability-zone **Link**

Interface	Method	API
Server Group Operations	create_server_group(self, **attrs)	POST /v2/{project_id}/os-server-groups **Link**
	delete_server_group(self, server_group, ignore_missing=True)	DELETE /v2/{project_id}/os-server-groups/{server_group_id} **Link**
	find_server_group(self, name_or_id, ignore_missing=True)	Name: GET /v2/{project_id}/os-server-groups **Link** Server_group_id: GET /v2/{project_id}/os-server-groups/{server_group_id} **Link**
	get_server_group(self, server_group)	GET /v2/{project_id}/os-server-groups/{server_group_id} **Link**
Quota Operations	get_limits(self)	GET /v2/{project_id}/limits **Link**
	query_quota(self, project_id)	GET /v2/{project_id}/os-quota-sets/{project_id} **Link**
	query_quota_default(self, project_id)	GET /v2/{project_id}/os-quota-sets/{project_id}/defaults **Link**
Volume Attachment Operations	delete_volume_attachment(self, volume_attachment, server, force_del=False, ignore_missing=True)	DELETE /v2/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id} **Link**
	volume_attachments(self, server)	GET /v2/{project_id}/servers/{server_id}/os-volume_attachments **Link**
	get_volume_attachment(self, volume_attachment, server)	GET /v2/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id} **Link**

Interface	Method	API
	create_volume_attachment(self, server, **attrs)	POST /v2/{project_id}/servers/{server_id}/os-volume_attachments **Link**
Metadata Operations	get_server_metadata(self, server, key=None)	• Key=None: GET /v2/{project_id}/servers/{server_id}/metadata **Link** • Key !=None: GET /v2/{project_id}/servers/{server_id}/metadata/{key} **Link**
	update_server_metadata(self, server, key, value)	PUT /v2/{project_id}/servers/{server_id}/metadata/{key} **Link**
	set_server_metadata(self, server, **metadata)	POST /v2/{project_id}/servers/{server_id}/metadata **Link**
	delete_server_metadata(self, server, keys)	DELETE /v2/{project_id}/servers/{server_id}/metadata/{key} **Link**

The SDK interfaces based on the Nova v2.1 API are as follows. Invocation example: conn.ecs.create_server_ext()

Interface	Method	API
Server Interface Operations	get_vnc_address(headers=headers, server_id=server_id, **data)	POST /v2.1/{project_id}/servers/{server_id}/remote-consoles **Link**

The SDK interfaces based on the ECS v1.1 API are as follows.

Invocation example: conn.ecs.create_server_ext()

Interface	Method	API
Server Operations	create_server_ext(self, **data)	POST /v1.1/{project_id}/cloudservers **Link**
	resize_server_ext(self, server_id, **data)	POST /v1.1/{project_id}/cloudservers/{server_id}/resize **Link**

The SDK interfaces based on the ECS v1 API are as follows. Invocation example: conn.ecs.get_server()

Interface	Method	API
Server Operations	get_server(self, server_id)	GET /v1/{project_id}/cloudservers/{server_id} **Link**
	servers(self, paginated=True, **query)	GET /v1/{project_id}/cloudservers/detail{?flavor,name,status,limit,offset,not-tags,reservation_id,enterprise_project_id} **Link**
	def batch_change_os_server(self, **data)	POST/v1/{project_id}/cloudservers/{server_id}/changeos (Currently, no API link is available.)
	get_autorecovery(self, server_id)	GET /v1/{project_id}/cloudservers/{server_id}/autorecovery **Link**
	config_autorecovery(self, server_id, autorecovery)	PUT /v1/{project_id}/cloudservers/{server_id}/autorecovery **Link**
	reset_password(self, server_id, **data)	PUT /v1/{project_id}/cloudservers/{server_id}/os-reset-password **Link**

Interface	Method	API
Flavor Operations	flavors(self, **query)	GET /v1/{project_id}/cloudservers/flavors{?availability_zone} **Link**
Quota Operations	quotas(self)	GET /v1/{project_id}/cloudservers/limits **Link**
Server Tag Operations	create_server_tags(self, server_id, **data)	POST /v1/{project_id}/cloudservers/{server_id}/tags/action **Link**
	delete_server_tags(self, server_id, **data)	POST /v1/{project_id}/cloudservers/{server_id}/tags/action **Link**
	get_server_tags(self, server_id)	GET /v1/{project_id}/cloudservers/{server_id}/tags **Link**
	get_project_tags(self)	GET /v1/{project_id}/cloudservers/tags **Link**
Server Interface Operations	get_vnc_address(server_id, **data)	POST /v1/{project_id}/cloudservers/{server_id}/remote_console **Link**

6.2.5 EVS

The SDK interfaces based on the EVS v2 API are as follows. Invocation example: conn.evs.create_volume_()

Interface	Method	API
Volume Operations	create_volume(**data)	POST /v2/{project_id}/cloudvolumes **Link**
	update_volume(volume_id, **data)	PUT /v2/{project_id}/cloudvolumes/{volume_id} **Link**

Interface	Method	API
	get_volume(volume_id)	GET /v2/{project_id}/cloudvolumes/{volume_id} **Link**
	resize_volume(self, volume_id, **data)	POST /v2/{project_id}/cloudvolumes/{volume_id}/action **Link**
	volumes(self, **query)	GET /v2/{project_id}/cloudvolumes/detail **Link**
Job Operations	get_job(self, job_id)	GET /v1/{project_id}/jobs/{job_id} **Link**

The SDK interfaces based on the EVS v2.1 API are as follows.

Invocation example: conn.evs.create_volume_ext()

Interface	Method	API
Volume Operations	create_volume_ext(self, **attrs)	POST /v2.1/{project_id}/cloudvolumes **Link**
	resize_volume_ext(self, volume_id, **data)	POST /v2.1/{project_id}/cloudvolumes/{volume_id}/action **Link**

The SDK interfaces based on the Cinder v2 API are as follows.

Invocation example: conn.block_store.create_volume()

Interface	Method	API
Snapshot Operations	get_snapshot(self, snapshot)	GET /v2/{project_id}/snapshots/{snapshot_id} **Link**
	snapshots(self, details=True, **query)	GET /v2/{project_id}/snapshots/detail **Link**

Interface	Method	API
	create_snapshot(self, **attrs)	POST /v2/{project_id}/snapshots **Link**
	delete_snapshot(self, snapshot, ignore_missing=True)	DELETE /v2/{project_id}/snapshots/{snapshot_id} **Link**
Type Operations	get_type(self, type)	GET /v2/{project_id}/types/{volume_type_id} **Link**
	types(self)	GET /v2/{project_id}/types **Link**
Volume Operations	get_volume(self, volume)	GET /v2/{project_id}/volumes/{volume_id} **Link**
	volumes(self, details=True, **query)	GET /v2/{project_id}/volumes/detail **Link**
	create_volume(self, **attrs)	POST /v2/{project_id}/volumes **Link**
	delete_volume(self, volume, ignore_missing=True, cascade=False)	DELETE /v2/{project_id}/volumes/{volume_id} **Link**

6.2.6 AS

The SDK interfaces based on the AS v1 API are as follows. Invocation example: conn.auto_scaling.create_group()

Interface	Method	API
Group Operations	create_group(self, **attrs)	POST /autoscaling-api/v1/{project_id}/scaling_group **Link**
	groups(self, **query)	GET /autoscaling-api/v1/{project_id}/scaling_group **Link**

Interface	Method	API
	get_group(self, group)	GET /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id} **Link**
	update_group(self, group, **attrs)	PUT /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id} **Link**
	delete_group(self, group, ignore_missing=True)	DELETE /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id} **Link**
	resume_group(self, group)	POST /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id}/action **Link**
	pause_group(self, group)	POST /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id}/action **Link**
Config Operations	create_config(self, name, **attrs)	POST /autoscaling-api/v1/{project_id}/scaling_configuration **Link**
	configs(self, **query)	GET /autoscaling-api/v1/{project_id}/scaling_configuration **Link**
	get_config(self, config)	GET /autoscaling-api/v1/{project_id}/scaling_configuration/{scaling_configuration_id} **Link**
	delete_config(self, config, ignore_missing=True)	DELETE /autoscaling-api/v1/{project_id}/scaling_configuration/{scaling_configuration_id} **Link**

Interface	Method	API
	batch_delete_configs(self, configs)	POST /autoscaling-api/v1/{project_id}/scaling_configurations **Link**
Instance Operations	instances(self, group, **query)	GET /autoscaling-api/v1/{project_id}/scaling_group_instance/{scaling_group_id}/list **Link**
	remove_instance(self, instance, delete_instance=False, ignore_missing=True)	DELETE /autoscaling-api/v1/{project_id}/scaling_group_instance/{instance_id} **Link**
	batch_add_instances(self, group, instances)	POST /autoscaling-api/v1/{project_id}/scaling_group_instance/{scaling_group_id}/action **Link**
	batch_remove_instances(self, group, instances, delete_instance=False)	POST /autoscaling-api/v1/{project_id}/scaling_group_instance/{scaling_group_id}/action **Link**
Policy Operations	create_policy(self, **attrs)	POST /autoscaling-api/v1/{project_id}/scaling_policy **Link**
	update_policy(self, policy, **attrs)	PUT /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**
	policies(self, group, **query)	GET /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_group_id}/list **Link**
	get_policy(self, policy)	GET /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**

Interface	Method	API
	execute_policy(self, policy)	POST /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id}/action **Link**
	resume_policy(self, policy)	POST /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id}/action **Link**
	pause_policy(self, policy)	POST /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id}/action **Link**
	delete_policy(self, policy, ignore_missing=True)	DELETE /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**
Activity Operations	activities(self, group, **query)	GET /autoscaling-api/v1/{project_id}/scaling_activity_log/{scaling_group_id} **Link**
Quota Operations	quotas(self, group=None)	GET /autoscaling-api/v1/{project_id}/quotas **Link**
	quotas(self, group=None)	GET /autoscaling-api/v1/{project_id}/quotas/{scaling_group_id} **Link**
Lifecycle_hook Operations	create_lifecycle_hook(self, group, **attrs)	POST /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id} **Link**
	lifecycle_hooks(self, group)	GET /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/list **Link**

Interface	Method	API
	get_lifecycle_hook(self, group, lifecycle_hook)	GET /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
	update_lifecycle_hook(self, group, lifecycle_hook, **attrs)	PUT /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
	delete_lifecycle_hook(self, group, lifecycle_hook)	DELETE /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
	call_back_instance(self, group, **attrs)	PUT /autoscaling-api/v1/{project_id}/scaling_instance_hook/{scaling_group_id}/callback **Link**
	get_group_hanging_instance(self, group, **query)	GET /autoscaling-api/v1/{project_id}/scaling_instance_hook/{scaling_group_id}/list{?instance_id} **Link**
Notification Operations	create_notification(self, group, **data)	PUT /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id} **Link**
	notifications(self, group)	GET /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id} **Link**

Interface	Method	API
	delete_notification(self, group, topic)	DELETE /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id}/{topic_urn} Link

6.2.7 CES

The SDK interfaces based on the CES v1.0 API are as follows. Invocation example: conn.cloud_eye.metrics()

Interface	Method	API
Metric Operations	metrics(self, **query)	GET /V1.0/{project_id}/metrics
Alarm Operations	alarms(self, **query)	GET /V1.0/{project_id}/alarms
	get_alarm(self, alarm)	GET /V1.0/{project_id}/alarms/{alarm_id}
	enable_alarm(self, alarm) disable_alarm(self, alarm)	PUT /V1.0/{project_id}/alarms/{alarm_id}/action
	delete_alarm(self, alarm, ignore_missing=True)	DELETE /V1.0/{project_id}/alarms/{alarm_id}
Metric Data Operations	metric_aggregations(self, **query)	GET /V1.0/{project_id}/metric-data
	add_metric_data(self, data)	POST /V1.0/{project_id}/metric-data
Quota Operations	quotas(self)	GET /V1.0/{project_id}/quotas

6.2.8 DNS

The SDK interfaces based on the DNS v2 API are as follows. Invocation example: conn.dns.create_zone()

Interface	Method	API
Zone Operations	create_zone(self, **attrs)	POST /v2/zones **Link** (public network) **Link** (intranet)
	get_zone(self, zone)	GET /v2/zones/{zone_id} **Link** (public network) **Link** (intranet)
	zones(self, **query)	GET /v2/zones **Link** (public network) **Link** (intranet)
	delete_zone(self, zone, ignore_missing=True)	DELETE /v2/zones/{zone_id} **Link** (public network) **Link** (intranet)
	nameservers(self, zone)	GET /v2/zones/{zone_id}/nameservers **Link** (public network) **Link** (intranet)
	add_router_to_zone(self, zone, **router)	POST /v2/zones/{zone_id}/associaterouter **Link**
	remove_router_from_zone(self, zone, **router)	POST /v2/zones/{zone_id}/disassociaterouter **Link**
Recordset Operations	create_recordset(self, zone, **attrs)	POST /v2/zones/{zone_id}/recordsets **Link**
	get_recordset(self, zone, recordset)	GET /v2/zone/{zone_id}/recordsets/{recordset_id} **Link**
	all_recordsets(self, **query)	GET /v2/recordsets **Link**
	recordsets(self, zone, **query)	GET /v2/zones/{zone_id}/recordsets **Link**
	delete_recordset(self, zone, recordset, ignore_missing=True)	DELETE /v2/zones/{zone_id}/recordsets/{recordset_id} **Link**

Interface	Method	API
PTR Record Operations	create_ptr(self, **attrs)	PATCH /v2/reverse/floatingips/{region}:{floatingip_id} **Link**
	restore_ptr(self, region, floating_ip_id)	PATCH /v2/reverse/floatingips/{region}:{floatingip_id} **Link**
	ptrs(self, **query)	GET /v2/reverse/floatingips **Link**
	get_ptr(self, region, floating_ip_id)	GET /v2/reverse/floatingips/{region}:{floatingip_id} **Link**

6.2.9 ELB

The SDK interfaces based on the ELB v2.0 API are as follows. Invocation example: conn.network.loadbalancers()

Interface	Method	API
Loadbalancer Operations	loadbalancers(self, **query)	GET /v2.0/lbaas/loadbalancers **Link**
	get_loadbalancer(self, lb)	GET /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	create_loadbalancer(self, **attrs)	POST /v2.0/lbaas/loadbalancers **Link**
	update_loadbalancer(self, lb, **attrs)	PUT /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	delete_loadbalancer(self, lb, ignore_missing=True)	DELETE /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	get_loadbalancer_status_stree(self, lb)	GET /v2.0/lbaas/loadbalancers/{loadbalancer_id}/statuses **Link**
Listener Operations	listeners(self, **query)	GET /v2.0/lbaas/listeners **Link**

Interface	Method	API
	get_listener(self, lsn)	GET /v2.0/lbaas/listeners/{listener_id} **Link**
	create_listener(self, **attrs)	POST /v2.0/lbaas/listeners **Link**
	update_listener(self, lsn, **attrs)	PUT /v2.0/lbaas/listeners/{listener_id} **Link**
	delete_listener(self, lsn, ignore_missing=True)	DELETE /v2.0/lbaas/listeners/{listener_id} **Link**
Pool Operations	pools(self, **query)	GET /v2.0/lbaas/pools **Link**
	get_pool(self, pol)	GET /v2.0/lbaas/pools/{pool_id} **Link**
	create_pool(self, **attrs)	POST /v2.0/lbaas/pools **Link**
	update_pool(self, pol, **attrs)	PUT /v2.0/lbaas/pools/{pool_id} **Link**
	delete_pool(self, pol, ignore_missing=True)	DELETE /v2.0/lbaas/pools/{pool_id} **Link**
Member Operations	members(self, **query)	GET /v2.0/lbaas/pools/{pool_id}/members **Link**
	get_member(self, mem, pool = None)	GET /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**
	create_member(self, **attrs)	POST /v2.0/lbaas/pools/{pool_id}/members **Link**
	update_member(self, mem, **attrs)	PUT /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**
	delete_member(self, mem, pool = None, ignore_missing=True)	DELETE /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**

Interface	Method	API
Healthmonitor Operations	healthmonitors(self, **query)	GET /v2.0/lbaas/healthmonitors Link
	get_healthmonitor(self, hlth)	GET /v2.0/lbaas/healthmonitors/{healthmonitor_id} Link
	create_healthmonitor(self, **attrs)	POST /v2.0/lbaas/healthmonitors Link
	update_healthmonitor(self, hlth, **attrs)	PUT /v2.0/lbaas/healthmonitors/{healthmonitor_id} Link
	delete_healthmonitor(self, hlth, ignore_missing=True)	DELETE /v2.0/lbaas/healthmonitors/{healthmonitor_id} Link
Policy Operations	poliycies(self, **query)	GET /v2.0/lbaas/l7policies Link
	get_policy(self, plc)	GET /v2.0/lbaas/l7policies/{l7policy_id} Link
	create_policy(self, **attrs)	POST /v2.0/lbaas/l7policies Link
	update_policy(self, plc, **attrs)	PUT /v2.0/lbaas/l7policies/{l7policy_id} Link
	delete_policy(self, plc, ignore_missing=True)	DELETE /v2.0/lbaas/l7policies/{l7policy_id} Link
Rule Operations	rules(self, **query)	GET /v2.0/lbaas/l7policies/{l7policy_id}/rules Link
	get_rule(self, rul, policy)	GET /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} Link
	create_rule(self, **attrs)	POST /v2.0/lbaas/l7policies/{l7policy_id}/rules Link
	update_rule(self, rul, **attrs)	PUT /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} Link

Interface	Method	API
	delete_rule(self, rul, policy, ignore_missing=True)	DELETE /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} **Link**
Whitelist Operations	whitelists(self, **query)	GET /v2.0/lbaas/whitelists **Link**
	get_whitelist(self, wl)	GET /v2.0/lbaas/whitelists/{whitelist_id} **Link**
	create_whitelist(self, **attrs)	POST /v2.0/lbaas/whitelists **Link**
	update_whitelist(self, wl, **attrs)	PUT /v2.0/lbaas/whitelists/{whitelist_id} **Link**
	delete_whitelist(self, wl, ignore_missing=True)	DELETE /v2.0/lbaas/whitelists/{whitelist_id} **Link**
Certificate Operations	certificates(self, **query)	GET /v2.0/lbaas/certificates **Link**
	get_certificate(self, cf)	GET /v2.0/lbaas/certificates/{certificate_id} **Link**
	create_certificate(self, **attrs)	POST /v2.0/lbaas/certificates **Link**
	update_certificate(self, cf, **attrs)	PUT /v2.0/lbaas/certificates/{certificate_id} **Link**
	delete_certificate(self, cf, ignore_missing=True)	DELETE /v2.0/lbaas/certificates/{certificate_id} **Link**

The SDK interfaces based on the ELB v1.0 API are as follows. Invocation example: conn.load_balancer.create_load_balancer()

Interface	Method	API
LoadBalancer Operations	create_load_balancer(self, **attrs)	POST /v1.0/{project_id}/elbaas/loadbalancers **Link**

Interface	Method	API
	get_load_balancer(self, load_balancer)	GET /v1.0/{project_id}/elbaas/loadbalancers/{loadbalancer_id} **Link**
	load_balancers(self, **query)	GET /v1.0/{project_id}/elbaas/loadbalancers **Link**
	update_load_balancer(self, load_balancer, **attrs)	PUT /v1.0/{project_id}/elbaas/loadbalancers/{loadbalancer_id} **Link**
	delete_load_balancer(self, load_balancer, ignore_missing=True)	DELETE /v1.0/{project_id}/elbaas/loadbalancers/{loadbalancer_id} **Link**
Listener Operations	create_listener(self, **attrs)	POST /v1.0/{project_id}/elbaas/listeners **Link**
	get_listener(self, listener)	GET /v1.0/{project_id}/elbaas/listeners/{listener_id} **Link**
	listeners(self, **query)	GET /v1.0/{project_id}/elbaas/listeners?loadbalancer_id={loadbalancer_id} **Link**
	update_listener(self, listener, **attrs)	PUT /v1.0/{project_id}/elbaas/listeners/{listener_id} **Link**
	delete_listener(self, listener, ignore_missing=True)	DELETE /v1.0/{project_id}/elbaas/listeners/{listener_id} **Link**
HealthCheck Operations	create_health_check(self, **attrs)	POST /v1.0/{project_id}/elbaas/healthcheck **Link**
	get_health_check(self, health_check)	GET /v1.0/{project_id}/elbaas/healthcheck/{healthcheck_id} **Link**

Interface	Method	API
	update_health_check(self, health_check, **attrs)	PUT /v1.0/{project_id}/elbaas/healthcheck/{healthcheck_id} **Link**
	delete_health_check(self, health_check, ignore_missing=True)	DELETE /v1.0/{project_id}/elbaas/healthcheck/{healthcheck_id} **Link**
Member Operations	add_members_to_listener(self, listener, members)	POST /v1.0/{project_id}/elbaas/listeners/{listener_id}/members **Link**
	remove_members_of_listener(self, listener, members)	POST /v1.0/{project_id}/elbaas/listeners/{listener_id}/members/action **Link**
	listener_members(self, listener, **query)	GET /v1.0/{project_id}/elbaas/listeners/{listener_id}/members **Link**
Certificate Operations	create_certificate(self, **attrs)	POST /v1.0/{project_id}/elbaas/certificate **Link**
	certificates(self)	GET /v1.0/{project_id}/elbaas/certificate **Link**
	update_certificate(self, certificate, **attrs)	PUT /v1.0/{project_id}/elbaas/certificate/{certificate_id} **Link**
	delete_certificate(self, certificate, ignore_missing=True)	DELETE /v1.0/{project_id}/elbaas/certificate/{certificate_id} **Link**

6.2.10 VBS

The SDK interfaces based on the VBS v2 API are as follows. Invocation example: conn.volume_backup.create_backup()

Interface	Method	API
VolumeBackup Operations	create_backup(**backup)	POST /v2/{project_id}/cloudbackups
	create_native_backup(**backup)	Post /v2/{project_id}/backups
	restore_backup(volume_backup_id, volume_id)	POST/v2/{project_id}/cloudbackups/{backup_id}/restore
	backups(self, details=False, **query)	GET /v2/{project_id}/backups
	get_backup(self, backup)	GET /v2/{project_id}/backups/{backup_id}
	delete_backup(self, backup, ignore_missing=True)	DELETE /v2/{project_id}/backups/{backup_id}
	get_job(self, job)	GET /v1/{project_id}/jobs/{job_id}
VolumeBackupPolicy Operations	create_backup_policy(volume_backup_name, **data)	POST /v2/{project_id}/backuppolicy
	backup_policies()	GET /v2/{project_id}/backuppolicy
	update_backup_policy(policy, **updated)	PUT /v2/{project_id}/backuppolicy/{policy_id}
	delete_backup_policy(policy)	DELETE /v2/{project_id}/backuppolicy/{policy_id}
	link_resources_to_policy(policy, volumes)	POST /v2/{project_id}/backuppolicyresources
	unlink_resources_of_policy(policy, volumes)	POST /v2/{project_id}/backuppolicyresources/{policy_id}/deleted_resources
	execute_policy(policy)	POST /v2/{project_id}/backuppolicy/{policy_id}/action
	tasks(backup_policy_id, **query)	GET /v2/{project_id}/backuppolicy/{policy_id}/backuptasks
	enable_policy(policy)	PUT /v2/{project_id}/backuppolicy/{policy_id}
	disable_policy(policy)	PUT /v2/{project_id}/backuppolicy/{policy_id}

6.2.11 KMS

The SDK interfaces based on the KMS v1.0 API are as follows. Invocation example: conn.kms.create_key()

Interface	Method	API
Key Operations	create_key(**kwargs)	POST /v1.0/{project_id}/kms/create-key
	keys(**query)	POST /v1.0/{project_id}/kms/list-keys
	describe_key(key, **kwargs)	POST /v1.0/{project_id}/kms/describe-key
	disable_key(key, **params)	POST /v1.0/{project_id}/kms/disable-key
	enable_key(key, **params)	POST /v1.0/{project_id}/kms/enable-key
	schedule_deletion_key(key, pending_days, **params)	POST /v1.0/{project_id}/kms/schedule-key-deletion
	cancel_deletion_key(key, **params)	POST /v1.0/{project_id}/kms/cancel-key-deletion
random Operations	gen_random(**params)	POST /v1.0/{project_id}/kms/gen-random
DataKey Operations	create_datakey(key, **params)	POST /v1.0/{project_id}/kms/create-datakey
	create_datakey_wo_plain(key, **params)	POST /v1.0/{project_id}/kms/create-datakey-without-plaintext
	encrypt_datakey(datakey, **params)	POST /v1.0/{project_id}/kms/encrypt-datakey
	decrypt_datakey(datakey, **params)	POST /v1.0/{project_id}/kms/decrypt-datakey
InstanceNumber Operations	get_instance_number()	GET /v1.0/{project_id}/kms/user-instances
Quota Operations	get_quota()	GET /v1.0/{project_id}/kms/user-quotas

6.2.12 Anti-DDoS

The SDK interfaces based on Anti-DDoS v1 APIs are as follows. Invocation example: conn.anti_ddos.query_config_list()

Interface	Method	API
QueryConfigList Operations	query_config_list()	GET /v1/{project_id}/antiddos/query_config_list **Link**
EIP Operations	get_floating_ip(floating_ip)	GET /v1/{project_id}/antiddos/{floating_ip_id} **Link**
	update_floating_ip(floating_ip, **attrs)	PUT /v1/{project_id}/antiddos/{floating_ip_id} **Link**
	query_task_status(task_id)	GET /v1/{project_id}/query_task_status **Link**
	floating_ips(**query)	GET /v1/{project_id}/antiddos **Link**
	get_eip_status(floating_ip_id)	GET /v1/{project_id}/antiddos/{floating_ip_id}/status **Link**
	list_eip_daily(floating_ip_id, **query)	GET /v1/{project_id}/antiddos/{floating_ip_id}/daily **Link**
	list_eip_log(floating_ip_id, **query)	GET /v1/{project_id}/antiddos/{floating_ip_id}/logs **Link**
EIPWeekly Operations	get_eip_weekly(period_start_date)	GET /v1/{project_id}/antiddos/weekly **Link**
AlertConfig Operations	get_alert_config()	GET /v1/{project_id}/warnalert/alertconfig/query **Link**

6.2.13 DMS

The SDK interfaces based on the DMS v1.0 API are as follows. Invocation example: conn.dms.create_queue()

Interface	method	API
Queue Operations	create_queue(**kwargs)	POST /v1.0/{project_id}/queues
	queues()	GET /v1.0/{project_id}/queues
	get_queue(queue)	GET /v1.0/{project_id}/queues/{queue_id}
	delete_queue(queue, ignore_missing=True)	DELETE /v1.0/{project_id}/queues/{queue_id}
Group Operations	create_groups(queue, **kwargs)	POST /v1.0/{project_id}/queues/{queue_id}/groups
	groups(queue)	GET /v1.0/{project_id}/queues/{queue_id}/groups
	delete_group(queue, group)	DELETE /v1.0/{project_id}/queues/{queue_id}/groups/{consumer_group_id}
Message Operations	send_messages(queue, **kwargs)	POST /v1.0/{project_id}/queues/{queue_id}/messages
	consume_message(queue, consume_group, **query)	GET /v1.0/{project_id}/queues/{queue_id}/groups/{consumer_group_id}/messages
	ack_consumed_message(consumed_message, status='success')	POST /v1.0/{project_id}/queues/{queue_id}/groups/{consumer_group_id}/ack
Quota Operations	quotas()	GET /v1.0/{project_id}/quotas/dms

6.2.14 MRS

The SDK interfaces based on the MRS v1.1 API are as follows. Invocation example: conn.map_reduce.create_cluster_and_run_job()

Interface	Method	API
Cluster Operations	create_cluster_and_run_job(**cluster,**job)	POST /v1.1/{project_id}/run-job-flow
	expand_cluster("cluster-id", expand_node_amount)	PUT /v1.1/{project_id}/cluster_infos/{cluster_id}
	reduce_cluster("cluster-id", reduce_node_amount, includes=includes, excludes=excludes)	PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

Interface	Method	API
	get_cluster("cluster-id")	GET /v1.1/{project_id}/cluster_infos/{cluster_id}
	delete_cluster("cluster-id")	DELETE /v1.1/{project_id}/clusters/{cluster_id}
Job Operations	exe_job(**exe)	POST /v1.1/{project_id}/jobs/submit-job
	job_exes(**query)	GET /v1.1/{project_id}/job-exes
	get_job_exe("job-exe-id")	GET /v1.1/{project_id}/job-exes/{job_exe_id}
JobExecution Operations	delete_job_execution("job-execution-id")	DELETE /v1.1/{project_id}/job-executions/{job_execution_id}

6.2.15 RDS

The SDK interfaces based on the RDS v3 API are as follows. Invocation example: **conn.rds_v3.create_instance()**.

Interface	Method	API
Instance Operations	create_instance(self, **kwargs)	POST /v3/{project_id}/instances **Link**
	instances(self)	GET /v3/{project_id}/instances?id={id}&name={name}&type={type}&datastore_type={datastore_type}&vpc_id={vpc_id}&subnet_id={subnet_id}&offset={offset}&limit={limit} **Link**
	delete_instance(self, **kwargs)	DELETE /v3/{project_id}/instances/{instance_id} **Link**
	resize_instance(self, instance, flavorRef)	POST /v3/{project_id}/instances/{instance_id}/action **Link**
	resize_instance_volume(instance, size)	POST /v3/{project_id}/instances/{instance_id}/action **Link**

Interface	Method	API
	restart_instance(instance)	POST /v3/{project_id}/instances/{instance_id}/action **Link**
	single_to_ha(instance,**single_to_ha_param)	POST /v3/{project_id}/instances/{instance_id}/action **Link**
Backup Operations	create_backup(self, **kwarg)	POST /v3/{project_id}/backups **Link**
	backups(self, **kwarg)	GET /v3/{project_id}/backups?instance_id={instance_id}&backup_id={backup_id}&backup_type={backup_type}&offset={offset}&limit={limit}&begin_time={begin_time}&end_time={end_time} **Link**
	restore_time(self, **kwarg)	GET v3/{project_id}/instances/{instance_id}/restore-time **Link**
	delete_backup(self, id)	DELETE /v3/{project_id}/backups/{backup_id} **Link**
	backup_files(**query)	GET /v3/{project_id}/backup-files?backup_id={backup_id} **Link**
	recovery_instance(**recovery_point)	POST /v3/{project_id}/instances **Link**
Backup Policy Operations	create_backup_policy(self, **kwargs)	PUT /v3/{project_id}/instances/{instance_id}/backups/policy **Link**
	get_backup_policy(self, **kwargs)	GET /v3/{project_id}/instances/{instance_id}/backups/policy **Link**
Flavor Operations	flavors(self, **kwargs)	GET /v3/{project_id}/flavors/{database_name}?version_name={version_name} **Link**
Datastore Operations	datastore_versions(self, dbname)	GET /v3/{project_id}/datastores/{database_name} **Link**

Interface	Method	API
Configuration Operations	configurations(self, **kwargs)	GET /v3/{project_id}/configurations Link
Log Operations	list_instance_errorlog(**query)	GET instances/{instance_id}/errorlog?start_date={start_date}&end_date={end_date} Link
	list_instance_slowlog(**query)	GET /v3/{project_id}/instances/{instance_id}/slowlog?start_date={start_date}&end_date={end_date} Link

6.2.16 CDN

The SDK interfaces based on the CDN v1.0 API are as follows. Invocation example: conn.cdn.domains()

Interface	Method	API
Acceleration domain name Operations	domains(enterprise_project_id='ALL', page_size=100, page_number=1, **query)	GET /v1.0/cdn/domains
	create_domain(**attrs)	POST /v1.0/cdn/domains
	get_domain(domain_id)	GET /v1.0/cdn/domains/{domain_id}/detail
	get_domain_detail_by_enterprise_project_id(domain_id, enterprise_project_id)	GET /v1.0/cdn/domains/{domain_id}/detail
	delete_domain(domain_id), ignore_missing=True)	DELETE /v1.0/cdn/domains/{domain_id}
	delete_domain_by_enterprise_project_id(domain_id, enterprise_project_id)	DELETE /v1.0/cdn/domains/{domain_id}
	enable_domain(domain_id)	PUT /v1.0/cdn/domains/{domain_id}/enable

Interface	Method	API
	enable_domain_by_enterprise_project_id(domain_id, enterprise_project_id)	PUT /v1.0/cdn/domains/{domain_id}/enable
	disable_domain(domain_id)	PUT /v1.0/cdn/domains/{domain_id}/disable
	disable_domain_by_enterprise_project_id(domain_id, enterprise_project_id)	PUT /v1.0/cdn/domains/{domain_id}/disable
	set_domain_sources(domain_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/origin
	set_domain_sources_by_enterprise_project_id(domain_id, enterprise_project_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/origin
	set_domain_origin_host(domain_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/originhost
	set_domain_origin_host_by_enterprise_project_id(domain_id, enterprise_project_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/originhost
	get_domain_origin_host(domain_id)	GET /v1.0/cdn/domains/{domain_id}/originhost
	get_domain_origin_host_by_enterprise_project_id(domain_id, enterprise_project_id)	GET /v1.0/cdn/domains/{domain_id}/originhost
	set_domain_range_status(domain_id,**attrs)	PUT /v1.0/cdn/domains/{domainId}/range-switch
	set_domain_follow302_switch(domain_id,**attrs)	PUT /v1.0/cdn/domains/{domainId}/follow302-switch
	set_domain_referer(domain_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/referer
	set_domain_referer_by_enterprise_project_id(domain_id, enterprise_project_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/referer

Interface	Method	API
	get_domain_referer(domain_id)	GET /v1.0/cdn/domains/{domain_id}/referer
	get_domain_referer_by_enterprise_project_id(domain_id, enterprise_project_id)	GET /v1.0/cdn/domains/{domain_id}/referer
	get_domain_ip_acl(domain_id)	GET /v1.0/cdn/domains/{domainId}/ip-acl
	set_domain_ip_acl(domain_id,**attrs)	PUT /v1.0/cdn/domains/{domainId}/ip-acl
	set_domain_cache_rules(domain_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/cache
	set_domain_cache_rules_by_enterprise_project_id(domain_id, enterprise_project_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/cache
	get_domain_cache_rules(domain_id)	GET /v1.0/cdn/domains/{domain_id}/cache
	get_domain_cache_rules_by_enterprise_project_id(domain_id, enterprise_project_id)	GET /v1.0/cdn/domains/{domain_id}/cache
	set_domain_https(domain_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/https-info
	set_domain_https_by_enterprise_project_id(domain_id, enterprise_project_id, **attrs)	PUT /v1.0/cdn/domains/{domain_id}/https-info
	get_domain_https(domain_id)	GET /v1.0/cdn/domains/{domain_id}/https-info
	get_domain_https_by_enterprise_project_id(domain_id, enterprise_project_id)	GET /v1.0/cdn/domains/{domain_id}/https-info
	get_cdn_ips(ips)	GET /v1.0/cdn/ip-info
	get_cdn_ips_enterprise(ips, enterprise_project_id)	GET /v1.0/cdn/ip-info

Interface	Method	API
	set_domain_response_header(domain_id,**attrs)	PUT /v1.0/cdn/domains/{domainId}/response-header
	set_domain_response_header_enterprise(domain_id, enterprise_project_id,**attrs)	PUT /v1.0/cdn/domains/{domainId}/response-header
	get_domain_response_header(domain_id)	GET /v1.0/cdn/domains/{domainId}/response-header
	get_domain_response_header_enterprise(domain_id, enterprise_project_id)	GET /v1.0/cdn/domains/{domainId}/response-header
Statistic Operations	query_top_url(**query)	GET /v1.0/cdn/statistics/top-url
	query_domain_item_details(**query)	GET /v1.0/cdn/statistics/domain-item-details
	query_domain_item_location_details(**query)	GET /v1.0/cdn/statistics/domain-item-location-details
	query_network_traffic(**query)	GET /v1.0/cdn/statistics/flux
	query_network_traffic_detail(**query)	GET /v1.0/cdn/statistics/flux-detail
	query_bandwidth_peak(**query)	GET /v1.0/cdn/statistics/bandwidth
	query_bandwidth(**query)	GET /v1.0/cdn/statistics/bandwidth-detail
	query_summary(**query)	GET /v1.0/cdn/statistics/domain-summary
	query_summary_detail(**query)	GET /v1.0/cdn/statistics/domain-summary-detail
	summaries(**query)	GET /v1.0/cdn/statistics/domain
	query_region_detail_summary(**query)	GET /v1.0/cdn/statistics/region-detail-summary
	query_carrier_detail_summary(**query)	GET /v1.0/cdn/statistics/carrier-detail-summary

Interface	Method	API
	query_region_carrier_domain(**query)	GET /v1.0/cdn/statistics/region-carrier-domain
	query_region_carrier_detail(**query)	GET /v1.0/cdn/statistics/region-carrier-detail
Log Operations	logs(domain_name, query_date, page_size=100, page_number=1)	GET /v1.0/cdn/logs
	logs_by_enterprise_project_id (domain_name, query_date, page_size=100, page_number=1, enterprise_project_id)	GET /v1.0/cdn/logs
Preheating task Operations	create_preheat_task(**attrs)	POST /v1.0/cdn/preheatingtasks
Refreshing task Operations	create_refresh_task(**attrs)	POST /v1.0/cdn/refreshtasks
Query task Operations	tasks(page_size=100, page_number=1, **query)	GET /v1.0/cdn/historytasks
	get_task(task_id)	GET /v1.0/cdn/historytasks/{task_id}/detail

6.2.17 FGS

The SDK interfaces based on the FGS v2.0 Python API are as follows. Invocation example: conn.fgs.get_function_list().

Interface	method	API
Functions	functions(**function)	GET /v2/{project_id}/fgs/functions?marker={marker}&maxitems={maxitems}
	get_function_metadata(function_urn)	GET /v2/{project_id}/fgs/functions/{function_urn}/config
	get_function_code(function_urn)	GET /v2/{project_id}/fgs/functions/{function_urn}/code

	create_function(**attrs)	POST /v2/{project_id}/fgs/functions
	delete_function(function_urn)	DELETE /v2/{project_id}/fgs/functions/{function_urn}
	update_function_code(function_urn, **attrs)	PUT /v2/{project_id}/fgs/functions/{function_urn}/code
	update_function_metadata(function_urn, **attrs)	PUT /v2/{project_id}/fgs/functions/{function_urn}/config
	publish_function_version(function_urn,**attrs)	POST /v2/{project_id}/fgs/functions/{function_urn}/versions
	get_function_version(function_urn,**attrs)	GET /v2/{project_id}/fgs/functions/{function_urn}/versions?marker={marker}&maxitems={maxitems}
	create_function_aliase(function_urn,**attrs)	POST /v2/{project_id}/fgs/functions/{function_urn}/aliases
	update_function_aliase(function_urn,alias_name, **attrs)	PUT /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
	delete_function_aliase(function_urn,alias_name)	DELETE /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
	get_function_aliase(function_urn,alias_name)	GET /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
	function_aliases(function_urn, **function)	GET /v2/{project_id}/fgs/functions/{function_urn}/aliases
	execute_function_synchronously(function_urn,**attrs)	POST /v2/{project_id}/fgs/functions/{function_urn}/invocations
	execute_function_asynchronously(function_urn,**attrs)	POST /v2/{project_id}/fgs/functions/{function_urn}/invocations-async
triggers	triggers(function_urn,**attrs)	GET /v2/{project_id}/fgs/triggers/{function_urn}
	get_trigger(function_urn,trigger_type_code,trigger_id)	GET /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
	delete_all_triggers(function_urn)	DELETE /v2/{project_id}/fgs/triggers/{function_urn}

	create_trigger(function_urn,**attrs)	POST /v2/{project_id}/fgs/triggers/{function_urn}
	delete_trigger(function_urn,trigger_type_code,trigger_id)	DELETE /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}

6.2.18 TMS

The SDK interfaces based on TMS v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
PredefineTag	create_predefine_tag (**attrs)	POST /v1.0/predefine_tags/action **Link**
	list_predefine_tags(**query)	GET/v1.0/predefine_tags **Link**
	update_predefine_tag (**attrs)	PUT /v1.0/predefine_tags **Link**

6.2.19 EPS

The SDK interfaces based on EPS v1.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
EnterpriseProject	create_enterprise_project(**attrs)	POST /v1.0/enterprise-projects **Link**
	list_enterprise_projects(**query)	GET /v1.0/enterprise-projects **Link**
	get_enterprise_project(enterpriseProjectId)	GET /v1.0/enterprise-projects/{id} **Link**
	update_enterprise_project(enterpriseProjectId, **attrs)	PUT /v1.0/enterprise-projects/{id} **Link**

Interface	Method	API
	enterprise_project_quotas	GET /v1.0//enterprise-projects/quotas **Link**
	operate_enterprise_project(enterpriseProjectId, **attrs)	POST/v1.0/enterprise-projects/{id}/action **Link**
	filter_resource_enterprise_project(enterpriseProjectId, **attrs)	POST /v1.0/enterprise-projects/{id}/resources/filter **Link**
	migrate_resource_enterprise_project(enterpriseProjectId, **attrs)	POST /v1.0/enterprise-projects/{id} /resources-migrate **Link**

6.3 Go

6.3.1 ECS

The SDK interfaces based on the ECS v1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
CloudServers	GetJobResult(client *gophercloud.ServiceClient, id string)	GET /v1/{project_id}/jobs/{job_id} **Link**
	GetServerRecoveryStatus(client *gophercloud.ServiceClient, serverID string)	GET /v1/{project_id}/cloudservers/{server_id}/autorecovery **Link**
	ConfigServerRecovery(client *gophercloud.ServiceClient, serverID string, opts string)	PUT /v1/{project_id}/cloudservers/{server_id}/autorecovery **Link**
	Get(client *gophercloud.ServiceClient, serverID string)	GET /v1/{project_id}/cloudservers/{server_id} **Link**

Interface	Method	API
	ListDetail(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/{project_id}/cloudservers/detail{?flavor,name,status,limit,offset,not-tags,reservation_id,enterprise_project_id,tags} **Link**
	BatchStart(client *gophercloud.ServiceClient, opts BatchStartOpts)	POST /v1/{project_id}/cloudservers/action **Link**
	BatchReboot(client *gophercloud.ServiceClient, opts BatchRebootOpts)	POST /v1/{project_id}/cloudservers/action **Link**
	BatchStop(client *gophercloud.ServiceClient, opts BatchStopOpts)	POST /v1/{project_id}/cloudservers/action **Link**
	BatchUpdate(client *gophercloud.ServiceClient, opts BatchUpdateOpts)	PUT /v1/{project_id}/cloudservers/server-name **Link**
Tags	BatchCreateServer-Tags(client *gophercloud.ServiceClient, serverID string, opts BatchTagCreateOpts-Builder)	POST /v1/{project_id}/cloudservers/{server_id}/tags/action **Link**
	BatchDeleteServer-Tags(client *gophercloud.ServiceClient, serverID string, opts BatchTagDeleteOpts-Builder)	POST /v1/{project_id}/cloudservers/{server_id}/tags/action **Link**
	ListProjectTags(client *gophercloud.ServiceClient)	GET /v1/{project_id}/cloudservers/tags **Link**
	ListServerTags(client *gophercloud.ServiceClient, serverID string)	GET /v1/{project_id}/cloudservers/{server_id}/tags **Link**

The SDK interfaces based on the ECS v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
CloudServers	ReinstallOS(client *gophercloud.ServiceClient, serverID string, opts ReinstallOptsBuilder)	POST /v2/{project_id}/cloudservers/{server_id}/reinstallos **Link**

The SDK interfaces based on the Nova v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
Servers	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2/{project_id}/servers **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2/{project_id}/servers/detail **Link**
	Get(client *gophercloud.ServiceClient, id string)	GET /v2/{project_id}/servers/{server_id} **Link**
	Update(client *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PUT /v2/{project_id}/servers/{server_id} **Link**
	Delete(client *gophercloud.ServiceClient, id string)	DELETE /v2/{project_id}/servers/{server_id} **Link**
	Resize(client *gophercloud.ServiceClient, id string, opts ResizeOptsBuilder)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	ConfirmResize(client *gophercloud.ServiceClient, id string)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	RevertResize(client *gophercloud.ServiceClient, id string)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	Start(client *gophercloud.ServiceClient, id string)	POST /v2/{project_id}/servers/{server_id}/action **Link**

Interface	Method	API
	Stop(client *gophercloud.ServiceClient, id string)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	Create(client *gophercloud.ServiceClient, opts servers.CreateOptsBuilder)	POST /v2/{project_id}/os-volumes_boot **Link**
	Reboot(client *gophercloud.ServiceClient, id string, opts RebootOptsBuilder)	POST /v2/{project_id}/servers/{server_id}/action **Link**
	ResetMetadata(client *gophercloud.ServiceClient, id string, opts ResetMetadataOpts-Builder)	PUT /v2/{project_id}/servers/{server_id}/metadata **Link**
	Metadata(client *gophercloud.ServiceClient, id string)	GET /v2/{project_id}/servers/{server_id}/metadata **Link**
	UpdateMetadata(client *gophercloud.ServiceClient, id string, opts UpdateMetadataOpts-Builder)	POST /v2/{project_id}/servers/{server_id}/metadata **Link**
	Metadatum(client *gophercloud.ServiceClient, id, key string)	GET /v2/{project_id}/servers/{server_id}/metadata/{key} **Link**
	DeleteMetadatum(client *gophercloud.ServiceClient, id, key string)	DELETE /v2/{project_id}/servers/{server_id}/metadata/{key} **Link**
	ListInstanceActions(client *gophercloud.ServiceClient, serverID string)	GET /v2/{project_id}/servers/{server_id}/os-instance-actions **Link**
	GetInstanceActions(client *gophercloud.ServiceClient, serverID string, RequestID string)	GET /v2/{project_id}/servers/{server_id}/os-instance-actions/{request_id} **Link**

Interface	Method	API
	GetConsoleLog(client *gophercloud.ServiceClient, id string, length string)	POST /v2/{project_id}/servers/{server_id}/action **Link**
Interface	List(client *gophercloud.ServiceClient, serverID string)	GET /v2/{project_id}/servers/{server_id}/os-interface **Link**
	Get(client *gophercloud.ServiceClient, serverID, portID string)	GET /v2/{project_id}/servers/{server_id}/os-interface/{id} **Link**
	Create(client *gophercloud.ServiceClient, serverID string, opts CreateOptsBuilder)	POST /v2/{project_id}/servers/{server_id}/os-interface **Link**
	Delete(client *gophercloud.ServiceClient, serverID, portID string)	DELETE /v2/{project_id}/servers/{server_id}/os-interface/{id} **Link**
Flavors	ListDetail(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2/{project_id}/flavors/detail **Link**
	Get(client *gophercloud.ServiceClient, id string)	GET /v2/{project_id}/flavors/{flavor_id} **Link**
Images	ListDetail(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2/{project_id}/images/detail **Link**
	Delete(client *gophercloud.ServiceClient, id string)	DELETE /v2/{project_id}/images/{image_id} **Link**
	Get(client *gophercloud.ServiceClient, id string)	GET /v2/{project_id}/images/{image_id} **Link**
Key Pairs	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2/{project_id}/os-keypairs **Link**

Interface	Method	API
	Get(client *gophercloud.ServiceClient, name string)	GET /v2/{project_id}/os-keypairs/{keypair_name} **Link**
	Delete(client *gophercloud.ServiceClient, name string)	DELETE /v2/{project_id}/os-keypairs/{keypair_name} **Link**
	List(client *gophercloud.ServiceClient)	GET /v2/{project_id}/os-keypairs **Link**
Quotas	GetLimits(client *gophercloud.ServiceClient)	GET /v2/{project_id}/limits **Link**
	Get(client *gophercloud.ServiceClient, tenantID string)	GET /v2/{project_id}/os-quota-sets/{project_id} **Link**
	GetDefault(client *gophercloud.ServiceClient, ProjectID string)	GET /v2/{project_id}/os-quota-sets/{project_id}/defaults **Link**
Volumeattach	List(client *gophercloud.ServiceClient, serverID string)	GET /v2/{project_id}/servers/{server_id}/os-volume_attachments **Link**
	Create(client *gophercloud.ServiceClient, serverID string, opts CreateOptsBuilder)	POST /v2/{project_id}/servers/{server_id}/os-volume_attachments **Link**
	Get(client *gophercloud.ServiceClient, serverID, attachmentID string)	GET /v2/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id} **Link**
	Delete(client *gophercloud.ServiceClient, serverID, attachmentID string)	DELETE /v2/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id} **Link**

Interface	Method	API
	DeleteWithFlag(client *gophercloud.ServiceClient, serverID, volumeID string, deleteFlag int)	DELETE /v2/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id}?delete_flag={delete_flag} Link

6.3.2 EVS

The SDK interfaces based on the Cinder v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
Volumes	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2/{project_id}/volumes/detail Link
	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2/{project_id}/volumes Link
	Get(client *gophercloud.ServiceClient, id string)	GET /v2/{project_id}/volumes/{volume_id} Link
	Delete(client *gophercloud.ServiceClient, id string)	DELETE /v2/{project_id}/volumes/{volume_id} Link
	Update(client *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PUT /v2/{project_id}/volumes/{volume_id} Link

6.3.3 VPC

The SDK interfaces based on the VPC v1 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
Bandwidths	Get(client *gophercloud.ServiceClient, bandwidthId string)	GET /v1/{project_id}/bandwidths/{bandwidth_id} Link

Interface	Method	API
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/{project_id}/bandwidths **Link**
	Update(client *gophercloud.ServiceClient, bandwidthId string, opts UpdateOptsBuilder)	PUT /v1/{project_id}/bandwidths/{bandwidth_id} **Link**
Ports	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v1/{project_id}/ports **Link**
	Delete(client *gophercloud.ServiceClient, portId string)	DELETE /v1/{project_id}/ports/{port_id} **Link**
	Get(client *gophercloud.ServiceClient, portId string)	GET /v1/{project_id}/ports/{port_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/{project_id}/ports **Link**
	Update(client *gophercloud.ServiceClient, portId string, opts UpdateOptsBuilder)	PUT /v1/{project_id}/ports/{port_id} **Link**
PrivateIps	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v1/{project_id}/privateips **Link**
	Delete(client *gophercloud.ServiceClient, privateipId string)	DELETE /v1/{project_id}/privateips/{privateip_id} **Link**
	Get(client *gophercloud.ServiceClient, privateipId string)	GET /v1/{project_id}/privateips/{privateip_id} **Link**
	List(client *gophercloud.ServiceClient, subnetId string, opts ListOptsBuilder)	GET /v1/{project_id}/subnets/{subnet_id}/privateips **Link**
PublicIps	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v1/{project_id}/publicips **Link**

Interface	Method	API
	Delete(client *gophercloud.ServiceClient, publicipId string)	DELETE /v1/{project_id}/publicips/{publicip_id} **Link**
	Get(client *gophercloud.ServiceClient, publicipId string)	GET /v1/{project_id}/publicips/{publicip_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/{project_id}/publicips **Link**
	Update(client *gophercloud.ServiceClient, publicipId string, opts UpdateOptsBuilder)	PUT /v1/{project_id}/publicips/{publicip_id} **Link**
Quotas	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/{project_id}/quotas **Link**
SecurityGroupRules	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v1/security-group-rules **Link**
	Delete(client *gophercloud.ServiceClient, securityGroupsRulesId string)	DELETE /v1/security-group-rules/{security-groups-rules-id} **Link**
	Get(client *gophercloud.ServiceClient, securityGroupsRulesId string)	GET /v1/security-group-rules/{security-groups-rules-id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/security-group-rules **Link**
SecurityGroups	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v1/{project_id}/security-groups **Link**
	Delete(client *gophercloud.ServiceClient, securityGroupId string)	DELETE /v1/{project_id}/security-groups/{security_group_id} **Link**

Interface	Method	API
	Get(client *gophercloud.ServiceClient, securityGroupId string)	GET /v1/{project_id}/security-groups/{security_group_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/{project_id}/security-groups **Link**
Subnets	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v1/{project_id}/subnets **Link**
	Delete(client *gophercloud.ServiceClient, vpcId string, subnetId string)	DELETE /v1/{project_id}/vpcs/{vpc_id}/subnets/{subnet_id} **Link**
	Get(client *gophercloud.ServiceClient, subnetId string)	GET /v1/{project_id}/subnets/{subnet_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/{project_id}/subnets **Link**
	Update(client *gophercloud.ServiceClient, vpcId string, subnetId string, opts UpdateOptsBuilder)	PUT /v1/{project_id}/vpcs/{vpc_id}/subnets/{subnet_id} **Link**
Vpcs	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v1/{project_id}/vpcs **Link**
	Delete(client *gophercloud.ServiceClient, vpcId string)	DELETE /v1/{project_id}/vpcs/{vpc_id} **Link**
	Get(client *gophercloud.ServiceClient, vpcId string)	GET /v1/{project_id}/vpcs/{vpc_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v1/{project_id}/vpcs **Link**
	Update(client *gophercloud.ServiceClient, vpcId string, opts UpdateOptsBuilder)	PUT /v1/{project_id}/vpcs/{vpc_id} **Link**

The SDK interfaces based on the Neutron v2.0 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
Networks	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/networks **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/networks **Link**
	Get(client *gophercloud.ServiceClient, networkId string)	GET /v2.0/networks/{network_id} **Link**
	Update(client *gophercloud.ServiceClient, networkId string, opts UpdateOptsBuilder)	PUT /v2.0/networks/{network_id) **Link**
	Delete(client *gophercloud.ServiceClient, networkId string)	DELETE /v2.0/networks/{network_id} **Link**
Subnets	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/subnets **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/subnets **Link**
	Get(client *gophercloud.ServiceClient, subnetId string)	GET /v2.0/subnets/{subnet_id} **Link**
	Update(client *gophercloud.ServiceClient, subnetId string, opts UpdateOptsBuilder)	PUT /v2.0/subnets/{subnet_id} **Link**
	Delete(client *gophercloud.ServiceClient, subnetId string)	DELETE /v2.0/subnets/{subnet_id} **Link**
Ports	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/ports **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/ports **Link**

Interface	Method	API
	Get(client *gophercloud.ServiceClient, portId string)	GET /v2.0/ports/{port_id} **Link**
	Update(client *gophercloud.ServiceClient, portId string, opts UpdateOptsBuilder)	PUT /v2.0/ports/{port_id} **Link**
	Delete(client *gophercloud.ServiceClient, portId string)	DELETE /v2.0/ports/{port_id} **Link**
SecurityGroup	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/security-groups **Link**
	Update(client *gophercloud.ServiceClient, securityGroupId string, opts UpdateOptsBuilder)	PUT /v2.0/security-groups/{security_group_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOpts)	GET /v2.0/security-groups **Link**
	Get(client *gophercloud.ServiceClient, securityGroupId string)	GET /v2.0/security-groups/{security_group_id} **Link**
SecurityGroupRules	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/security-group-rules **Link**
	Delete(client *gophercloud.ServiceClient, securityGroupsRulesId string)	DELETE /v2.0/security-group-rules/{security-groups-rules-id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOpts)	GET /v2.0/security-group-rules **Link**
	Get(client *gophercloud.ServiceClient, securityGroupsRulesId string)	GET /v2.0/security-group-rules/{security-groups-rules-id} **Link**
Routers	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/routers **Link**

Interface	Method	API
	Delete(client *gophercloud.ServiceClient, routerId string)	DELETE /v2.0/routers/{router_id} **Link**
	Update(client *gophercloud.ServiceClient, routerId string, opts UpdateOptsBuilder)	PUT /v2.0/routers/{router_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOpts)	GET /v2.0/routers **Link**
	Get(client *gophercloud.ServiceClient, routerId string)	GET /v2.0/routers/{router_id} **Link**
	AddInterface(client *gophercloud.ServiceClient, routerId string, opts AddInterfaceOptsBuild-er)	PUT /v2.0/routers/{router_id}/add_router_interface **Link**
	RemoveInterface(client *gophercloud.ServiceClient, routerId string, opts RemoveInterfaceOpts-Builder)	PUT /v2.0/routers/{router_id}/remove_router_interface **Link**
FloatingIps	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/floatingips **Link**
	Delete(client *gophercloud.ServiceClient, floatingipId string)	DELETE /v2.0/floatingips/{floatingip_id} **Link**
	Update(client *gophercloud.ServiceClient, floatingipId string, opts UpdateOptsBuilder)	PUT /v2.0/floatingips/{floatingip_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOpts)	GET /v2.0/floatingips **Link**
	Get(client *gophercloud.ServiceClient, floatingipId string)	GET /v2.0/floatingips/{floatingip_id} **Link**

6.3.4 IAM

The SDK interfaces based on the Keystone v3 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
Tokens	Create(c *gophercloud.ServiceClient, opts AuthOptionsBuilder)	POST /v3/auth/tokens **Link**
	Validate(c *gophercloud.ServiceClient, token string)	GET /v3/auth/tokens **Link**
Service catalog	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v3/services **Link**
	Get(client *gophercloud.ServiceClient, serviceID string)	GET /v3/services/{service_id} **Link**
Endpoints	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v3/endpoints **Link**

6.3.5 IMS

The SDK interfaces based on the Glance v2 API are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
images	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2/images
	Delete(client *gophercloud.ServiceClient, id string)	DELETE /v2/images/{image_id}
	List(c *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2/images
	Update(client *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PATCH /v2/images/{image_id}
	Get(client *gophercloud.ServiceClient, id string)	GET /v2/images/{image_id}

6.3.6 ELB

The SDK interfaces based on the ELB v2 Go SDK are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
policy	func List(sc *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/lbaas/l7policies **Link**
	func Get(sc *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/l7policies/{policy_id} **Link**
	func Create(sc *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/lbaas/l7policies **Link**
	func Update(sc *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PUT /v2.0/lbaas/l7policies/{policy_id} **Link**
	func Delete(sc *gophercloud.ServiceClient, id string)	DELETE /v2.0/lbaas/l7policies/{policy_id} **Link**
certificate	func List(c *gophercloud.ServiceClient)	GET /v2.0/lbaas/certificates **Link**
	func Get(c *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/certificates/{certificate_id} **Link**
	func Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/lbaas/certificates **Link**
	func Update(c *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PUT /v2.0/lbaas/certificates/{certificate_id} **Link**
	func Delete(c *gophercloud.ServiceClient, id string)	DELETE /v2.0/lbaas/certificates/{certificate_id} **Link**
loadbalancer	func List(c *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/lbaas/loadbalancers **Link**

Interface	Method	API
	func Get(c *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	func Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/lbaas/loadbalancers **Link**
	func Update(c *gophercloud.ServiceClient, id string, opts UpdateOpts)	PUT /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	func Delete(c *gophercloud.ServiceClient, id string)	DELETE /v2.0/lbaas/loadbalancers/{loadbalancer_id} **Link**
	func GetStatuses(c *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/loadbalancers/{loadbalancer_id}/statuses **Link**
listener	func List(c *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/lbaas/listeners **Link**
	func Get(c *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/listeners/{listener_id} **Link**
	func Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/lbaas/listeners **Link**
	func Update(c *gophercloud.ServiceClient, id string, opts UpdateOpts)	PUT /v2.0/lbaas/listeners/{listener_id} **Link**
	func Delete(c *gophercloud.ServiceClient, id string) (r DeleteResult)	DELETE /v2.0/lbaas/listeners/{listener_id} **Link**
pool	func List(c *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/lbaas/pools **Link**
	func Get(c *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/pools/{pool_id} **Link**

Interface	Method	API
	func Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/lbaas/pools **Link**
	func Update(c *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PUT /v2.0/lbaas/pools/{pool_id} **Link**
	func Delete(c *gophercloud.ServiceClient, id string)	DELETE /v2.0/lbaas/pools/{pool_id} **Link**
Member	func ListMembers(c *gophercloud.ServiceClient, poolID string, opts ListMembersOptsBuilder)	GET /v2.0/lbaas/pools/{pool_id}/members **Link**
	func GetMember(c *gophercloud.ServiceClient, poolID string, memberID string)	GET /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**
	func CreateMember(c *gophercloud.ServiceClient, poolID string, opts CreateMemberOpts)	POST /v2.0/lbaas/pools/{pool_id}/members **Link**
	func UpdateMember(c *gophercloud.ServiceClient, poolID string, memberID string, opts UpdateMemberOptsBuilder)	PUT /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**
	func DeleteMember(c *gophercloud.ServiceClient, poolID string, memberID string)	DELETE /v2.0/lbaas/pools/{pool_id}/members/{member_id} **Link**
healthmonitor	func List(c *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/lbaas/healthmonitors **Link**
	func Get(c *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/healthmonitors/{healthmonitor_id} **Link**
	func Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/lbaas/healthmonitors **Link**

Interface	Method	API
	func Update(c *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PUT /v2.0/lbaas/healthmonitors/{healthmonitor_id} **Link**
	func Delete(c *gophercloud.ServiceClient, id string)	DELETE /v2.0/lbaas/healthmonitors/{healthmonitor_id} **Link**
whitelist	func List(c *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/lbaas/whitelists **Link**
	func Get(c *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/whitelists/{whitelist_id} **Link**
	func Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/lbaas/whitelists **Link**
	func Update(c *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PUT /v2.0/lbaas/whitelists/{whitelist_id} **Link**
	func Delete(c *gophercloud.ServiceClient, id string)	DELETE /v2.0/lbaas/whitelists/{whitelist_id} **Link**
rule	func List(c *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2.0/lbaas/l7policies/{l7policy_id}/rules **Link**
	func Get(c *gophercloud.ServiceClient, id string)	GET /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} **Link**
	func Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2.0/lbaas/l7policies/{l7policy_id}/rules **Link**
	func Update(c *gophercloud.ServiceClient, id string, opts UpdateOptsBuilder)	PUT /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} **Link**
	func Delete(c *gophercloud.ServiceClient, id string)	DELETE /v2.0/lbaas/l7policies/{l7policy_id}/rules/{l7rule_id} **Link**

6.3.7 AS

The SDK interfaces based on the AS v1 Go SDK are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
Configures	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /autoscaling-api/v1/{project_id}/scaling_configuration **Link**
	Delete(client *gophercloud.ServiceClient, scalingConfigurationId string)	DELETE /autoscaling-api/v1/{project_id}/scaling_configuration/{scaling_configuration_id} **Link**
	DeleteWithBatch(client *gophercloud.ServiceClient, opts DeleteWithBatchOptsBuilder)	POST /autoscaling-api/v1/{project_id}/scaling_configurations **Link**
	Get(client *gophercloud.ServiceClient, scalingConfigurationId string)	GET /autoscaling-api/v1/{project_id}/scaling_configuration/{scaling_configuration_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /autoscaling-api/v1/{project_id}/scaling_configuration **Link**
Groups	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /autoscaling-api/v1/{project_id}/scaling_group **Link**
	Delete(client *gophercloud.ServiceClient, scalingGroupId string, opts DeleteOptsBuilder)	DELETE /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id} **Link**
	Enable(client *gophercloud.ServiceClient, scalingGroupId string, opts EnableOptsBuilder)	POST /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id}/action **Link**

Interface	Method	API
	Get(client *gophercloud.ServiceClient, scalingGroupId string)	GET /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /autoscaling-api/v1/{project_id}/scaling_group **Link**
	Update(client *gophercloud.ServiceClient, scalingGroupId string, opts UpdateOptsBuilder)	PUT /autoscaling-api/v1/{project_id}/scaling_group/{scaling_group_id} **Link**
Instances	Action(client *gophercloud.ServiceClient, scalingGroupId string, opts ActionOptsBuilder)	POST /autoscaling-api/v1/{project_id}/scaling_group_instance/{scaling_group_id}/action **Link**
	Delete(client *gophercloud.ServiceClient, instanceId string, opts DeleteOptsBuilder)	DELETE /autoscaling-api/v1/{project_id}/scaling_group_instance/{instance_id} **Link**
	List(client *gophercloud.ServiceClient, scalingGroupId string, opts ListOptsBuilder)	GET /autoscaling-api/v1/{project_id}/scaling_group_instance/{scaling_group_id}/list **Link**
Lifecycle Hooks	CallBack(client *gophercloud.ServiceClient, scalingGroupId string, opts CallBackOptsBuilder)	PUT /autoscaling-api/v1/{project_id}/scaling_instance_hook/{scaling_group_id}/callback **Link**
	Create(client *gophercloud.ServiceClient,scalingGroupId string, opts CreateOptsBuilder)	POST /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id} **Link**

Interface	Method	API
	Delete(client *gophercloud.ServiceClient,scalingGroupId string,lifecycleHookName string)	DELETE /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
	Get(client *gophercloud.ServiceClient, scalingGroupId string, lifecycleHookName string)	GET /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
	List(client *gophercloud.ServiceClient, scalingGroupId string)	GET /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/list **Link**
	ListWithSuspension(client *gophercloud.ServiceClient, scalingGroupId string, opts ListWithSuspensionOptsBuilder)	GET /autoscaling-api/v1/{project_id}/scaling_instance_hook/{scaling_group_id}/list **Link**
	Update(client *gophercloud.ServiceClient, scalingGroupId string, lifecycleHookName string, opts UpdateOptsBuilder)	PUT /autoscaling-api/v1/{project_id}/scaling_lifecycle_hook/{scaling_group_id}/{lifecycle_hook_name} **Link**
Logs	List(client *gophercloud.ServiceClient, scalingGroupId string, opts ListOptsBuilder)	GET /autoscaling-api/v1/{project_id}/scaling_activity_log/{scaling_group_id} **Link**
Notifications	Delete(client *gophercloud.ServiceClient, scalingGroupId string, topicUrn string)	DELETE /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id}/{topic_urn} **Link**

Interface	Method	API
	ConfigNotification(client *gophercloud.ServiceClient, scalingGroupId string, opts ConfigNotificationOptsBuilder)	PUT /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id} **Link**
	List(client *gophercloud.ServiceClient, scalingGroupId string)	GET /autoscaling-api/v1/{project_id}/scaling_notification/{scaling_group_id} **Link**
Policies	Action(client *gophercloud.ServiceClient, scalingPolicyId string, opts ActionOptsBuilder)	POST /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id}/action **Link**
	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /autoscaling-api/v1/{project_id}/scaling_policy **Link**
	Delete(client *gophercloud.ServiceClient, scalingPolicyId string)	DELETE /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**
	Get(client *gophercloud.ServiceClient, scalingPolicyId string)	GET /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**
	List(client *gophercloud.ServiceClient, scalingGroupId string, opts ListOptsBuilder)	GET /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_group_id}/list **Link**
	Update (client *gophercloud.ServiceClient, scalingPolicyId string, opts UpdateOptsBuilder)	PUT /autoscaling-api/v1/{project_id}/scaling_policy/{scaling_policy_id} **Link**
PolicyLogs	List(client *gophercloud.ServiceClient, scalingPolicyId string, opts ListOptsBuilder)	GET /autoscaling-api/v1/{project_id}/scaling_policy_execute_log/{scaling_policy_id} **Link**

Interface	Method	API
Quotas	List(client *gophercloud.ServiceClient)	GET /autoscaling-api/v1/{project_id}/quotas **Link**
	ListWithInstances(client *gophercloud.ServiceClient, scalingGroupId string)	GET /autoscaling-api/v1/{project_id}/quotas/{scaling_group_id} **Link**
Tags	ListResourceTags(client *gophercloud.ServiceClient, resourceType string, resourceId string)	GET /autoscaling-api/v1/{project_id}/{resource_type}/{resource_id}/tags **Link**
	ListTenantTags(client *gophercloud.ServiceClient, resourceType string)	GET /autoscaling-api/v1/{project_id}/{resource_type}/tags **Link**
	ListInstanceTags(client *gophercloud.ServiceClient, resourceType string, opts InstanceOptsBuilder)	POST /autoscaling-api/v1/{project_id}/{resource_type}/resource_instances/action **Link**
	Update(client *gophercloud.ServiceClient, resourceType string, resourceId string, opts UpdateOptsBuilder)	POST /autoscaling-api/v1/{project_id}/{resource_type}/{resource_id}/tags/action **Link**

The SDK interfaces based on the AS v2.0 Go SDK are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
Policies	Create(client *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /autoscaling-api/v2/{project_id}/scaling_policy **Link**
	Get(client *gophercloud.ServiceClient, scalingPolicyId string)	GET /autoscaling-api/v2/{project_id}/scaling_policy/{scaling_policy_id} **Link**

Interface	Method	API
	GetPolicyListByResourceID(client *gophercloud.ServiceClient, scalingResourceId string, opts ResourceListOptsBuilder)	GET /autoscaling-api/v2/{project_id}/scaling_policy/{scaling_resource_id}/list **Link**
	Update(client *gophercloud.ServiceClient, scalingPolicyId string, opts UpdateOptsBuilder)	PUT /autoscaling-api/v2/{project_id}/scaling_policy/{scaling_policy_id} **Link**
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /autoscaling-api/v2/{project_id}/scaling_policy **Link**

6.3.8 FGS

The SDK interfaces based on the FGS v2 Go SDK are as follows. For details about the invoking methods, see the sample codes.

Resource	Method	API
Functions	Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder)	POST /v2/{project_id}/fgs/functions
	List(client *gophercloud.ServiceClient, opts ListOptsBuilder)	GET /v2/{project_id}/fgs/functions?{marker}=marker&{maxitems}=maxitems
	GetMetadata(c *gophercloud.ServiceClient, functionUrn string)	GET /v2/{project_id}/fgs/functions/{function_urn}/config
	GetCode(c *gophercloud.ServiceClient, functionUrn string)	GET /v2/{project_id}/fgs/functions/{function_urn}/code
	Delete(c *gophercloud.ServiceClient, functionUrn string)	DELETE /v2/{project_id}/fgs/functions/{function_urn}
	UpdateCode(c *gophercloud.ServiceClient, functionUrn string, opts UpdateOptsBuilder)	PUT /v2/{project_id}/fgs/functions/{function_urn}/code

Resource	Method	API
	UpdateMetadata(c *gophercloud.ServiceClient, functionUrn string, opts UpdateOptsBuilder)	PUT /v2/{project_id}/fgs/functions/{function_urn}/config
	CreateVersion(c *gophercloud.ServiceClient, opts CreateOptsBuilder, functionUrn string)	POST /v2/{project_id}/fgs/functions/{function_urn}/versions
	ListVersion(c *gophercloud.ServiceClient, opts ListOptsBuilder, functionUrn string)	GET /v2/{project_id}/fgs/functions/{function_urn}/versions?marker={marker}&maxitems={maxitems}
	CreateAlias(c *gophercloud.ServiceClient, opts CreateOptsBuilder, functionUrn string)	POST /v2/{project_id}/fgs/functions/{function_urn}/aliases
	UpdateAlias(c *gophercloud.ServiceClient, functionUrn, aliasName string, opts UpdateOptsBuilder)	PUT /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
	DeleteAlias(c *gophercloud.ServiceClient, functionUrn, aliasName string)	DELETE /v1.0/{project_id}/fss/functions/{function_urn}/aliases/{alias_name}
	GetAlias(c *gophercloud.ServiceClient, functionUrn, aliasName string)	GET /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
	ListAlias(c *gophercloud.ServiceClient, functionUrn string)	GET /v2/{project_id}/fgs/functions/{function_urn}/aliases
	Invoke(c *gophercloud.ServiceClient, m map[string]interface{}, functionUrn string)	POST /v2/{project_id}/fgs/functions/{function_urn}/invocations
	AsyncInvoke(c *gophercloud.ServiceClient, m map[string]interface{}, functionUrn string)	POST /v2/{project_id}/fgs/functions/{function_urn}/invocations-async

Resource	Method	API
Triggers	List(c *gophercloud.ServiceClient, functionUrn string)	GET /v2/{project_id}/fgs/triggers/{function_urn}
	Create(c *gophercloud.ServiceClient, opts CreateOptsBuilder, functionUrn string)	POST /v2/{project_id}/fgs/triggers/{function_urn}
	Delete(c *gophercloud.ServiceClient, functionUrn, triggerTypeCode, triggerId string)	DELETE /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
	Get(c *gophercloud.ServiceClient, functionUrn, triggerTypeCode, triggerId string)	GET /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
	DeleteAll(c *gophercloud.ServiceClient, functionUrn string)	DELETE /v2/{project_id}/fgs/triggers/{function_urn}

6.3.9 RDS

The SDK interfaces based on the RDS v3 Go SDK are as follows. For details about the invoking methods, see the sample codes.

Interface	Method	API
Datastores	func listURL(sc *gophercloud.ServiceClient, databasename string)	GET /v3/{project_id}/datastores/{database_name} **Link**
Instances	func createURL(sc *gophercloud.ServiceClient)	POST /v3/{project_id}/instances **Link**
	func resizeURL(sc *gophercloud.ServiceClient, instancesId string)	POST /v3/{project_id}/instances/{instance_id}/action **Link**
	func enlargeURL(sc *gophercloud.ServiceClient, instancesId string)	POST /v3/{project_id}/instances/{instance_id}/action **Link**

Interface	Method	API
	func singletohaURL(sc *gophercloud.ServiceClient, instancesId string)	POST /v3/{project_id}/instances/{instance_id}/action **Link**
	func restartURL(sc *gophercloud.ServiceClient, instancesId string)	POST /v3/{project_id}/instances/{instance_id}/action **Link**
	func deleteURL(sc *gophercloud.ServiceClient, serverID string)	DELETE /v3/{project_id}/instances/{instance_id} **Link**
	func listURL(sc *gophercloud.ServiceClient)	GET /v3/{project_id}/instances **Link**
	func listerrorlogURL(sc *gophercloud.ServiceClient, instanceID string)	GET /v3/{project_id}/instances/{instance_id}/errorlog **Link**
	func listslowlogURL(sc *gophercloud.ServiceClient, instanceID string)	GET /v3/{project_id}/instances/{instance_id}/slowlog **Link**
Configurations	func listURL(sc *gophercloud.ServiceClient)	GET /v3/{project_id}/configurations **Link**
	func createURL(sc *gophercloud.ServiceClient)	POST /v3/{project_id}/configurations **Link**
Backups	func updatepolicyURL(sc *gophercloud.ServiceClient, instanceId string)	PUT /v3/{project_id}/instances/{instance_id}/backups/policy **Link**
	func getpolicyURL(sc *gophercloud.ServiceClient, instanceID string)	GET /v3/{project_id}/instances/{instance_id}/backups/policy **Link**
	func createURL(sc *gophercloud.ServiceClient)	POST /v3/{project_id}/backups **Link**
	func listURL(sc *gophercloud.ServiceClient)	GET /v3/{project_id}/backups **Link**
	func listfilesURL(sc *gophercloud.ServiceClient)	GET /v3/{project_id}/backup-files **Link**

Interface	Method	API
	func deleteURL(sc *gophercloud.ServiceClient, backupId string)	DELETE /v3/{project_id}/backups/{backup_id} **Link**
	func getrestoretimeURL(sc *gophercloud.ServiceClient, instanceId string)	GET /v3/{project_id}/instances/{instance_id}/restore-time **Link**
	func restoreURL(sc *gophercloud.ServiceClient)	POST /v3/{project_id}/instances **Link**
	func recoveryURL(sc *gophercloud.ServiceClient)	GET /v3/{project_id}/instances/recovery **Link**
Database	func createURL(sc *gophercloud.ServiceClient, instanceID string)	POST /v3/{project_id}/instances/{instance_id}/database **Link**
	func listURL(sc *gophercloud.ServiceClient, instanceID string)	GET /v3/{project_id}/instances/{instance_id}/database/detail **Link**
	func deleteURL(sc *gophercloud.ServiceClient, instanceID string,dbName string)	DELETE /v3/{project_id}/instances/{instance_id}/database/{db_name} **Link**
	func createURL(sc *gophercloud.ServiceClient, instanceID string)	POST /v3/{project_id}/instances/{instance_id}/db_privilege **Link**
	func deleteURL(sc *gophercloud.ServiceClient, instanceID string)	DELETE /v3/{project_id}/instances/{instance_id}/db_privilege **Link**
DbUser	func createURL(sc *gophercloud.ServiceClient, instanceID string)	POST /v3/{project_id}/instances/{instance_id}/db_user **Link**
	func listURL(sc *gophercloud.ServiceClient, instanceID string)	GET /v3/{project_id}/instances/{instance_id}/db_user/detail **Link**
	func deleteURL(sc *gophercloud.ServiceClient, instanceID string, dbuser string)	DELETE v3/{project_id}/instances/{instance_id}/db_user/{user_name} **Link**

Interface	Method	API
Flavors	func listURL(sc *gophercloud.ServiceClient, databasename string)	GET /v3/{project_id}/flavors/{database_name} **Link**
StorageType	func listURL(sc *gophercloud.ServiceClient, databasename string)	GET/v3/{project_id}/storage-type/{database_name} **Link**

6.3.10 CES

Interface	Method	API
MetricService	List(client *gophercloud.ServiceClient, opts ListOptsBuilder) pagination.Pager	GET /V1.0/{project_id}/metrics **Link**
AlarmService	List(client *gophercloud.ServiceClient, opts ListOpts) pagination.Pager	GET /V1.0/{project_id}/alarms **Link**
	Get(client *gophercloud.ServiceClient, alarmId string) (r GetResult)	GET /V1.0/{project_id}/alarms/{alarm_id} **Link**
	Update(client *gophercloud.ServiceClient, alarmId string, opts UpdateOptsBuilder)	PUT /V1.0/{project_id}/alarms/{alarm_id}/action **Link**
	Update(client *gophercloud.ServiceClient, alarmId string, opts UpdateOptsBuilder)	PUT /V1.0/{project_id}/alarms/{alarm_id}/action **Link**
	Delete(client *gophercloud.ServiceClient, alarmId string) (r DeleteResult)	DELETE /V1.0/{project_id}/alarms/{alarm_id} **Link**
MetricDataService	Get(client *gophercloud.ServiceClient, opts GetOpts)	GET /V1.0/{project_id}/metric-data **Link**
	AddMetricData(client *gophercloud.ServiceClient, opts AddMetricDataOpts-Builder)	POST /V1.0/{project_id}/metric-data **Link**

Interface	Method	API
QuotaService	Get(client *gophercloud.ServiceClient) (r GetResult)	GET /V1.0/{project_id}/quotas **Link**

A Change History

Released On	Description
2020-05-18	This issue is the tenth official release, which incorporates the following changes: • Added the RDS Java SDK. • Deleted the RTS Java and Python SDKs.
2020-04-21	This issue is the ninth official release, which incorporates the following changes: • Updated IAM Python SDK. • Updated the links of TMS and EPS APIs.
2019-11-22	This issue is the eighth official release, which incorporates the following changes: • Added SDK of Python language for RDS v3. • Deleted SDK of Python language for RDS v1.
2019-11-15	This issue is the seventh official release, which incorporates the following changes: • Added SDK of Go language for RDS v3. • Added JAVA, Python, and Go SDKs of Function Graph Service (FGS).
2019-06-17	This issue is the sixth official release, which incorporates the following change: Updated examples of **IAM Java SDK**, **IAM Python SDK**, and **AS Python SDK**, and modified the display mode.
2019-05-16	This issue is the fifth official release, which incorporates the following change: • Updated the example of Go language in EVS and VPC services. Modified the display mode. • Added section **AS Go SDK User Guide** in the AS service.

Released On	Description
2019-04-11	This issue is the fourth official release, which incorporates the following change: Modified the display style of EVS and VPC Java and Python examples.
2018-11-15	This issue is the third official release, which incorporates the following change: Added SDKs of KMS and CDN.
2017-12-29	This issue is the second official release, which incorporates the following change: Added the SDKs for the Go language.
2017-11-29	This issue is the first official release.

