Alibaba Cloud Virtual Private Cloud

SDK Reference

Issue: 20190912

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use**

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified , reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates . The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd / d C :/ windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 SDK development guide

VPC supports Java, Python, Go, .NET, and PHP SDK development.

The following table provides the download links and development guides for the supported SDKs. For more information, see Alibaba Cloud SDK platform.

SDK	GitHub download address	Development guide
Java SDK	VPC Java SDK	Get started
Python SDK	VPC Python SDK	Get started
Go SDK	VPC Go SDK	Get started
.NET SDK	VPC .NET SDK	Get started
PHP SDK	VPC PHP SDK	Get started

2 Python SDK examples

2.1 Install Alibaba Cloud SDK for Python

This topic describes how to build a Python development environment on a local PC so you can run VPC Python SDK examples. Alibaba Cloud SDK for Python supports Python 2.7. To run VPC Python SDK examples, you first need to install the core library of Alibaba Cloud SDK for Python and then install VPC Python SDK. Python SDK supports Windows, Linux, and Mac operating systems, and all VPC Python SDK examples can run in a Windows system.

Procedure

1. Install Python.
   a) Download Python 2.7.

      Select and download the correct installation package from the download list. The package format should be `python-XYZ.msi` and XYZ represents the version number.

   b) Double-click the package to enter the Python installation wizard and use the default configurations directly.

   c) Set environment variables.

      Add the python installation path and the directories of the pip program in the `Path` row. Separate the directories with ;

      ```
      C:\Python27;C:\Python27\Scripts
      ```

   d) In the `cmd` command line, run the `python` command. If the following codes are displayed, you have entered the Python interactive environment and Python has been installed.

      ```
      Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AM
      D64)] on win32
      Type "help", "copyright", "credits" or "license" for more informatio n.
      >>>
      ```

2. Run the following command to install the core library of Alibaba Cloud SDK for Python.

```
pip install aliyun-python-sdk-core
```

For more information about Python and PIP, see Python document and PIP document.

3. Run the following command to install VPC SDK.

```
pip install aliyun-python-sdk-vpc
```

4. Run the following command to use `--user` to install SDK.

```
pip install --user aliyun-python-sdk-cms
```

2.2 Preparations

Before you run example VPC Python SDK files, you must complete the required configurations and ensure the following:

1. There is at least 14.88 USD in your account.
2. You have obtained an AccessKey. For more information, see #unique_12.
3. The VPC Python Example library is downloaded.
4. The *setup.py* directory is accessed and the following command is run to initiate the environment.

```
python setup.py install --user
```

2.3 Virtual Private Cloud (VPC)

2.3.1 Create and delete a VPC/VSwitch

This topic describes how to use the Alibaba Cloud Python SDK to create and delete a VPC and a VSwitch.

Context

In this tutorial, we will create a VPC in the China (Zhangjiakou) region and create a VSwitch under this VPC. Then we will delete the VPC and VSwitch.

Procedure

1. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* file.
2. Open the *consts . py* file in your text editor and configure the ACS_CLIENT parameter used for user authentication.
3. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ vpc* folder.
4. Open the *vpc_quick_ start . py* file in your text editor and configure your required parameters. Then, save the configurations and exit the editor.
5. Enter the directory of *vpc_quick_ start . py* and run the following command to run the example that creates and deletes the VPC and VSwitch.

```
# encoding = utf - 8
 import   sys
 import   json

 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   CreateVpcR
 equest
 from   aliyunsdkv  pc . request . v20160428   import   CreateVSwi
 tchRequest
 from   aliyunsdkv  pc . request . v20160428   import   DeleteVSwi
 tchRequest
 from   aliyunsdkv  pc . request . v20160428   import   DeleteVpcR
 equest
 from   aliyunsdkv  pc . request . v20160428   import   DescribeVS
 witchAttri  butesReque  st
 from   aliyunsdkv  pc . request . v20160428   import   DescribeVp
 cAttribute  Request
 from   sdk_lib . exception   import   ExceptionH  andler
 from   sdk_lib . check_stat  us   import   CheckStatu  s
 from   sdk_lib . consts   import  *
 from   sdk_lib . common_uti  l   import   CommonUtil

 class   VpcQuickSt  art ( object ):
     def   __init__ ( self ,  client ):
         self . client  =  client

     def   create_vpc ( self ):
        """
         create_vpc :  Create   a   VPC
         API   reference   link :  https :// www . alibabaclo  ud .
 com / help / doc - detail / 35737 . htm
        """
         try :
             request  =  CreateVpcR  equest . CreateVpcR  equest ()
             response  =  self . client . do_action_  with_excep
 tion ( request )
             response_j  son  =  json . loads ( response )
            #  Check   whether   the   VPC   is   in   the
 available   state
             if   CheckStatu  s . check_stat  us ( TIME_DEFAU
 LT_OUT ,  DEFAULT_TI  ME ,
```

```
                                        self . describe_v
pc_status ,
                                        AVAILABLE ,  response_j
son [' VpcId ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   delete_vpc ( self ,  params ):
        """
        delete_vpc :  Delete   a   VPC
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 35738 . htm
        """
        try :
            request  =  DeleteVpcR  equest . DeleteVpcR  equest ()
           #  The   ID   of   the   VPC   to   be   deleted
            request . set_VpcId ( params [' vpc_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_v  pc_attribu  te ( self ,  vpc_id ):
        """
        describe_v  pc_attribu  te :  Query   the   VPC   created
in   the   specified   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 94565 . htm
        """
        try :
            request  =  DescribeVp  cAttribute  Request .
DescribeVp  cAttribute  Request ()
           #  The   ID   of   the   VPC   to   be   queried
            request . set_VpcId ( vpc_id )
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_v  pc_status ( self ,  vpc_id ):
        """
        describe_v  pc_status :  Query   the   status   of   the
VPC   created   in   the   specified   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 94565 . htm
        """
        response  =  self . describe_v  pc_attribu  te ( vpc_id )
        return   response [" Status "]

    def   create_vsw  itch ( self ,  params ):
        """
        create_vsw  itch :  Create   a   VSwitch   instance
```

```
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 35745 . htm
       """
        try :
            request  =  CreateVSwi  tchRequest . CreateVSwi
tchRequest ()
           #  The   ID   of   the   zone   to   which   the
VSwitch   belongs .  To   query   the   region   ID ,  call   the
DescribeZo  nes   action .
            request . set_ZoneId ( params [' zone_id '])
           #  The   ID   of   the   VPC   to   which   the
VSwitch   belongs .
            request . set_VpcId ( params [' vpc_id '])
           #  The   CIDR   block   of   the   VSwitch
            request . set_CidrBl  ock ( params [' cidr_block '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
           #  Check   whether   the   VSwitch   is   in   the
available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                      self . describe_v
switch_sta  tus ,
                                      AVAILABLE ,  response_j
son [' VSwitchId ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_v  switch_att  ribute ( self ,  vswitch_id ):
       """
        describe_v  switch_att  ribute :  Query   the   status   of
  the   VSwitch   created   in   the   specified   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 94567 . htm
       """
        try :
            request  =  DescribeVS  witchAttri  butesReque  st .
DescribeVS  witchAttri  butesReque  st ()
            request . set_VSwitc  hId ( vswitch_id )
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_v  switch_sta  tus ( self ,  vswitch_id ):
       """
        describe_v  switch_sta  tus :  Query   the   status   of
the   VSwitch   created   in   the   specified   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 94567 . htm
       """
        response  =  self . describe_v  switch_att  ribute (
vswitch_id )
        return   response [" Status "]

    def   delete_vsw  itch ( self ,  params ):
```

```
        """
         delete_vsw  itch :  Delete   a   VSwitch   instance
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 35746 . htm
        """
         try :
             request  =  DeleteVSwi  tchRequest . DeleteVSwi
tchRequest ()
            #  The   ID   of   the   VSwitch   to   be   deleted
             request . set_VSwitc  hId ( params [' vswitch_id '])
             response  =  self . client . do_action_  with_excep
tion ( request )
             response_j  son  =  json . loads ( response )
            #  Check   if   the   VSwitch   has   been   deleted
             if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME  *  5 ,
                                          self . describe_v
switch_sta  tus ,
                                         '',  params [' vswitch_id
']):
                 return   response_j  son
         except   ServerExce  ption   as   e :
             ExceptionH  andler . server_exc  eption ( e )
         except   ClientExce  ption   as   e :
             ExceptionH  andler . client_exc  eption ( e )


def   main ():
    client  =  ACS_CLIENT

    vpc_quick_  start  =  VpcQuickSt  art ( client )

    params  = {}
    params [' zone_id '] = " cn - zhangjiako  u - b "
    params [' cidr_block '] = " 172 . 16 . 0 . 0 / 16 "

   #  Create   a   vpc
    vpc_json  =  vpc_quick_  start . create_vpc ()
    CommonUtil . log (" create_vpc ",  vpc_json )

   #  Create   a   vswitch
    params [' vpc_id '] =  vpc_json [' VpcId ']
    vswitch_js  on  =  vpc_quick_  start . create_vsw  itch (
params )
    CommonUtil . log (" create_vsw  itch ",  vswitch_js  on )

   #  Delete   the   vswitch
    params [' vswitch_id '] =  vswitch_js  on [' VSwitchId ']
    vswitch_js  on  =  vpc_quick_  start . delete_vsw  itch (
params )
    CommonUtil . log (" delete_vsw  itch ",  vswitch_js  on )

   #  Delete   the   vpc
    vpc_json  =  vpc_quick_  start . delete_vpc ( params )
    CommonUtil . log (" delete_vpc ",  vpc_json )


if   __name__  == " __main__ ":
```

```
    sys . exit ( main ())
```

```
python   vpc_quick_  start . py
```

The system displays the following output:

```
--------------------------- create_vpc ---------------------------
{
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RouteTable  Id ": " vtb - 8vbf9ud7xr  cn9xxxxxxx  x ",
  " VRouterId ": " vrt - 8vb1qjnxcm  03nxxxxxxx  x ",
  " VpcId ": " vpc - 8vb67v4ozd  8wfxxxxxxx  x ",
  " RequestId ": " 5052F988 - 75CC - 46AD - A1A6 - 0E9E445BD0  D5 "
}
--------------------------- create_vsw  itch
 ---------------------------
{
  " VSwitchId ": " vsw - 8vbqn2at0k  ljjxxxxxxx  x ",
  " RequestId ": " 0BA1ABF7 - 21CF - 4460 - 9A86 - 0BB783886E  58 "
}
--------------------------- delete_vsw  itch
 ---------------------------
{
  " RequestId ": " D691F04B - A6EE - 49A7 - A434 - 4A45DD3AA0  B8 "
}
--------------------------- delete_vpc ---------------------------
{
  " RequestId ": " 4570F816 - AB8D - 45EA - 8913 - 6AE787C163  2C "
}
```

2.3.2 Create a custom route table

This topic describes how to use the Alibaba Cloud Python SDK to create a custom route table.

Context

To create a custom route table, the procedure described in this tutorial is as follows:

1. Create a VPC in the China (Zhangjiakou) region.
2. Create a VSwitch under the VPC.
3. Create a custom route table named as sdk_route_table.
4. Query the created VSwitch.
5. Associate the route table with the VSwitch in the same VPC.
6. Dissociate the route table from the VSwitch in the same VPC.
7. Delete the custom route table.
8. Delete the VSwitch.
9. Delete the VPC.

Procedure

1. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* folder.
2. Open the *consts . py* file in your text editor and configure the ACS_CLIENT parameter used for user authentication.
3. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ vpc* folder.
4. Open the *vpc_quick_ start . py* file in your text editor and configure your required parameters. Then, save the configurations and exit the editor.

```
# encoding = utf - 8
 import   sys
 import   json
 import   time

 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   CreateRout
 eEntryRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DeleteRout
 eEntryRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   AssociateE
 ipAddressR  equest
 from   aliyunsdkv  pc . request . v20160428   import   Unassociat
 eEipAddres  sRequest
 from   aliyunsdkv  pc . request . v20160428   import   DescribeRo
 uteTablesR  equest
 from   sdk_lib . common_uti  l   import   CommonUtil
 from   sdk_lib . sdk_vpc   import   Vpc
 from   sdk_lib . sdk_vswitc  h   import   VSwitch
 from   sdk_lib . exception   import   ExceptionH  andler
 from   sdk_lib . check_stat  us   import   CheckStatu  s
 from   sdk_lib . consts   import  *

 class   RouteTable ( object ):
     def   __init__ ( self ,  client ):
         self . client  =  client


     def   create_rou  te_table ( self ,  params ):
        """
         create_rou  te_table :
         API   reference   link :  https :// www . alibabaclo  ud .
 com / help / doc - detail / 87586 . htm
        """
         try :
             request  =  CreateRout  eEntryRequ  est . CreateRout
 eEntryRequ  est ()
             request . set_action  _name (" CreateRout  eTable ")
            #  The   ID   of   the   VPC   to   which   the   custom
   route   table   belongs .
             request . add_query_  param (" VpcId ",  params ['
 vpc_id '])
            #  The   name   of   the   route   table .
             request . add_query_  param (" RouteTable  Name ",
 params [' route_tabl  e_name '])
```

```
            response = self.client.do_action_with_exception(request)
            response_json = json.loads(response)
            route_table_id = response_json['RouteTableId']
            # Check if the route table is in the available state.
            if CheckStatus.check_status(TIME_DEFAULT_OUT, DEFAULT_TIME * 5,
                                        self.describe_route_table_status,
                                        route_table_id, route_table_id):
                return response_json
        except ServerException as e:
            ExceptionHandler.server_exception(e)
        except ClientException as e:
            ExceptionHandler.client_exception(e)

    def associate_route_table(self, params):
        """
        associate_route_table: Associate the custom route table with the VSwitch in the same VPC
        API reference link: https://www.alibabacloud.com/help/doc-detail/87599.htm
        """
        try:
            request = AssociateEipAddressRequest.AssociateEipAddressRequest()
            request.set_action_name("AssociateRouteTable")
            # The ID of the route table
            request.add_query_param("RouteTableId", params['route_table_id'])
            # The ID of the VSwitch
            request.add_query_param("VSwitchId", params['vswitch_id'])
            response = self.client.do_action_with_exception(request)
            response_json = json.loads(response)
            time.sleep(DEFAULT_TIME * 5)
            return response_json
        except ServerException as e:
            ExceptionHandler.server_exception(e)
        except ClientException as e:
            ExceptionHandler.client_exception(e)

    def unassociate_route_table(self, params):
        """
        unassociate_route_table: Dissociate the route table from the VSwitch
        API reference link: https://www.alibabacloud.com/help/doc-detail/87628.htm
        """
        try:
            request = UnassociateEipAddressRequest.UnassociateEipAddressRequest()
            request.set_action_name("UnassociateRouteTable")
            # The ID of the route table
            request.add_query_param("RouteTableId", params['route_table_id'])
            # The ID of the VSwitch
            request.add_query_param("VSwitchId", params['vswitch_id'])
```

```
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            time . sleep ( DEFAULT_TI  ME  *  5 )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   delete_rou  te_table ( self ,  params ):
        """
        delete_rou  te_table :  Delete   the   custom   route
table
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 87601 . htm
        """
        try :
            request  =  DeleteRout  eEntryRequ  est . DeleteRout
eEntryRequ  est ()
            request . set_action  _name (" DeleteRout  eTable ")
           #  The   ID   of   the   route   table
            request . add_query_  param (" RouteTable  Id ",
params [' route_tabl  e_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
           #  Check   if   the   route   table   has   been
deleted
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME  *  5 ,
                                        self . describe_r
oute_table  _status ,
                                       '',  params [' route_tabl
e_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_r  oute_table ( self ,  route_tabl  e_id ):
        """
        describe_r  oute_table :  Query   the   route   table
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36014 . htm
        """
        try :
            request  =  DescribeRo  uteTablesR  equest .
DescribeRo  uteTablesR  equest ()
           #  The   ID   of   the   route   table
            request . set_RouteT  ableId ( route_tabl  e_id )
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_r  oute_table  _status ( self ,  route_tabl
e_id ):
        """
```

```
        describe_r  oute_table  _status :  Query   the   status
of   the   route   table
        """
        response  =  self . describe_r  oute_table ( route_tabl
e_id )
        if   len ( response [" RouteTable  s "][" RouteTable "])
==  0 :
            return  ''
        return   response [" RouteTable  s "][" RouteTable "][ 0 ]
[" RouteTable  Id "]

def   main ():
    client  =  ACS_CLIENT

    vpc  =  Vpc ( client )
    vswitch  =  VSwitch ( client )
    route_tabl  e  =  RouteTable ( client )

    params  = {}
    params [' cidr_block '] = " 172 . 16 . 0 . 0 / 16 "
    params [' zone_id '] = " cn - zhangjiako  u - b "
    params [' route_tabl  e_name '] = " sdk_route_  table "

   # Create   a   vpc
    vpc_json  =  vpc . create_vpc ()
    CommonUtil . log (" create_vpc ",  vpc_json )

   # Create   a   vswitch
    params [' vpc_id '] =  vpc_json [' VpcId ']
    vswitch_js  on  =  vswitch . create_vsw  itch ( params )
    CommonUtil . log (" create_vsw  itch ",  vswitch_js  on )

   # Create   a   route   table
    route_tabl  e_json  =  route_tabl  e . create_rou  te_table (
params )
    CommonUtil . log (" create_rou  te_table ",  route_tabl  e_json
)

   # Query   the   vswitch
    params [' vswitch_id '] =  vswitch_js  on [' VSwitchId ']
    vswitch_js  on  =  vswitch . describe_v  switch_att  ribute (
params [' vswitch_id '])
    CommonUtil . log (" describe_v  switch_att  ribute ",
vswitch_js  on )

   # Associate   the   route   table   with   the   vswitch
    params [' route_tabl  e_id '] =  route_tabl  e_json ['
RouteTable  Id ']
    associate_  json  =  route_tabl  e . associate_  route_tabl  e
( params )
    CommonUtil . log (" associate_  route_tabl  e ",  associate_
json )

   # Dissociate   the   route   table   from   the   vswitch
    unassociat  e_json  =  route_tabl  e . unassociat  e_route_ta
ble ( params )
    CommonUtil . log (" unassociat  e_route_ta  ble ",  unassociat
e_json )

   # Delete   the   route   table
    delete_rou  te_table_j  son  =  route_tabl  e . delete_rou
te_table ( params )
    CommonUtil . log (" delete_rou  te_table ",  delete_rou
te_table_j  son )
```

```
    # Delete   the   vswitch
     delete_vsw  itch_json  =  vswitch . delete_vsw  itch ( params )
     CommonUtil . log (" delete_vsw  itch ",  delete_vsw  itch_json
 )

    # Delete   the   vpc
     delete_vpc  _json  =  vpc . delete_vpc ( params )
     CommonUtil . log (" delete_vpc ",  delete_vpc  _json )

 if   __name__  == " __main__ ":
     sys . exit ( main ())
```

5. **Enter the** *vpc_route_ table . py* **directory and run the following command to create the custom route entry.**

```
 python   vpc_route_  table . py
```

The system displays the following output:

```
--------------------------- create_vpc --------------------------
{
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RouteTable  Id ": " vtb - 8vb65a5hqy  8pcxxxxxxx  x ",
  " VRouterId ": " vrt - 8vbbbiftzi  zc3xxxxxxx  x ",
  " VpcId ": " vpc - 8vbebihln0  01gxxxxxxx  x ",
  " RequestId ": " 862F279B - 4A27 - 4300 - 87A1 - 047FB9961A  F2 "
}

--------------------------- create_vsw  itch
 ---------------------------
{
  " VSwitchId ": " vsw - 8vb30klhn2  is5xxxxxxx  x ",
  " RequestId ": " 1DA17173 - CB61 - 4DCE - 9C29 - AABFDF3001  A6 "
}

--------------------------- create_rou  te_table
 ---------------------------
{
  " RouteTable  Id ": " vtb - 8vbc4iwpo1  3apxxxxxxx  x ",
  " RequestId ": " 01E66E67 - 7801 - 4705 - A02A - 853BA7EEA8  9F "
}

--------------------------- describe_v  switch_att  ribute
 ---------------------------

{
  " Status ": " Available ",
  " NetworkAcl  Id ": "",
  " VpcId ": " vpc - 8vbebihln0  01gxxxxxxx  x ",
  " Descriptio  n ": "",
  " RouteTable ": {
    " RouteTable  Id ": " vtb - 8vb65a5hqy  8pcxxxxxxx  x ",
    " RouteTable  Type ": " System "
  },
  " CidrBlock ": " 172 . 16 . 0 . 0 / 16 ",
  " CreationTi  me ": " 2019 - 04 - 12T03 : 08 : 43Z ",
  " CloudResou  rces ": {
    " CloudResou  rceSetType ": []
  },
  " ZoneId ": " cn - zhangjiako  u - b ",
```

```
  " ResourceGr  oupId ": " rg - acfmxazbxx  xxxxxx ",
  " VSwitchId ": " vsw - 8vb30klhn2  is5xxxxxx  x ",
  " RequestId ": " C5A20BA3 - E998 - 498D - 8900 - 35AE5FDFFB  77 ",
  " Ipv6CidrBl  ock ": "",
  " VSwitchNam  e ": "",
  " AvailableI  pAddressCo  unt ":  252 ,
  " IsDefault ":  false
}

--------------------------- associate_  route_tabl  e
 ---------------------------
{
  " RequestId ": " 5FC0143B - D34B - 47DC - 8D49 - AFD222EA58  76 "
}

--------------------------- unassociat  e_route_ta  ble
 ---------------------------
{
  " RequestId ": " F0194718 - 6E4C - 496C - 9DA8 - 1B88DF1D6F  AD "
}

--------------------------- delete_rou  te_table
 ---------------------------
{
  " RequestId ": " B5C068A6 - 137C - 4337 - 8E3A - 9E30E17267  03 "
}

--------------------------- delete_vsw  itch
 ---------------------------
{
  " RequestId ": " 26DEDBF8 - 2F0D - 4A13 - 8CB3 - 23A84C9477  04 "
}

--------------------------- delete_vpc ---------------------------
{
  " RequestId ": " E1B2641F - 5911 - 40E4 - 9F36 - CC0B2EDD17  47 "
}
```

2.3.3 Create a custom route entry

This topic describes how to use the Alibaba Cloud Python SDK to create a custom route entry on a VRouter or VBR.

Prerequisites

Before you can create a custom route entry, you need to ensure that:

- **A VPC and a VSwitch are created in the China (Zhangjiakou) region.**
- **The next hop for the custom route entry is created in the China (Zhangjiakou) region. The next hop can be an ECS instance, an HAVIP, a VPN Gateway, a NAT Gateway, an ENI, or a router interface.**

Context

The detailed steps used in this tutorial to create a custom route entry are as follows:

1. Create a custom route table named sdk_route_table in the China (Zhangjiakou) region.
2. Query the VSwitches under the VPC vpc-8vb7ztbjqomi9xxxxxxxx.
3. Associate the custom route table with the VSwitch vsw-8vbfqpcijj0d1xxxxxxxx.
4. Add a custom route entry to the custom route table. The destination CIDR block of the entry is 168.168.0.0/16, the type of the next hop is ECS instance, and the ID of the next hop is i-8vbgsnt7046axxxxxxxx.
5. Delete the route entry.
6. Disassociate the custom route table from the VSwitch.
7. Delete the custom route table.

Procedure

1. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* folder.
2. Open the *consts . py* file in your text editor and configure the `ACS_CLIENT` parameter used for user authentication.
3. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ vpc* folder.
4. Open the *vpc_route_ entry . py* file in your editor and configure your required parameters. Then, save the configurations and exit the editor.

```
# encoding = utf - 8
 import   sys
 import   json
 import   time

 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   CreateRout
 eEntryRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DeleteRout
 eEntryRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DescribeRo
 uteTablesR  equest
 from   sdk_lib . exception   import   ExceptionH  andler
 from   sdk_lib . check_stat  us   import   CheckStatu  s
 from   sdk_lib . common_uti  l   import   CommonUtil
 from   sdk_lib . sdk_vswitc  h   import   VSwitch
 from   sdk_lib . sdk_route_  table   import   RouteTable
 from   sdk_lib . consts   import  *

 class   RouteEntry ( object ):
     def   __init__ ( self ,  client ):
         self . client  =  client

     def   create_rou  te_entry ( self ,  params ):
```

```
        """
         create_rou  te_entry :  Create   a   route   entry
         API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36012 . htm
        """
         try :
             request  =  CreateRout  eEntryRequ  est . CreateRout
eEntryRequ  est ()
            #  The   ID   of   the   route   table
             request . set_RouteT  ableId ( params [' route_tabl
e_id '])
            #  The   destinatio  n   CIDR   block   of   the
custom   route   entry
             request . set_Destin  ationCidrB  lock ( params ['
destinatio  n_cidr_blo  ck '])
            #  The   type   of   the   next   hop
             request . set_NextHo  pType ( params [' nexthop_ty  pe
'])
            #  The   ID   of   the   instance   used   as   the
next   hop
             request . set_NextHo  pId ( params [' nexthop_id '])
             response  =  self . client . do_action_  with_excep
tion ( request )
             response_j  son  =  json . loads ( response )
            #  Check   if   the   route   entry   is   in   the
available   state
             if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                          self . describe_r
oute_entry  _status ,
                                          AVAILABLE ,  params ['
route_tabl  e_id ']):
                 return   response_j  son
         except   ServerExce  ption   as   e :
             ExceptionH  andler . server_exc  eption ( e )
         except   ClientExce  ption   as   e :
             ExceptionH  andler . client_exc  eption ( e )

    def   delete_rou  te_entry ( self ,  params ):
        """
         delete_rou  te_entry :  Delete   the   route   entry
          API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36013 . htm
        """
         try :
             request  =  DeleteRout  eEntryRequ  est . DeleteRout
eEntryRequ  est ()
            #  The   ID   of   the   route   table   to   which
the   route   entry   belongs
             request . set_RouteT  ableId ( params [' route_tabl
e_id '])
            #  The   destinatio  n   CIDR   block   of   the   route
  entry
             request . set_Destin  ationCidrB  lock ( params ['
destinatio  n_cidr_blo  ck '])
            #  The   ID   of   the   instance   used   as   the
next   hop
             request . set_NextHo  pId ( params [' nexthop_id '])
             response  =  self . client . do_action_  with_excep
tion ( request )
             response_j  son  =  json . loads ( response )
             time . sleep ( DEFAULT_TI  ME )
             return   response_j  son
         except   ServerExce  ption   as   e :
```

```
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_r  oute_entry  _vrouter ( self ,  params ):
        """
        describe_r  oute_entry  _vrouter :  Query   the   route
entry
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36014 . htm
        """
        try :
            request  =  DescribeRo  uteTablesR  equest .
DescribeRo  uteTablesR  equest ()
           #  The   ID   of   the   VRouter   or   VBR   to   which
  the   route   table   belongs
            request . set_VRoute  rId ( params [' vrouter_id '])
           #  The   ID   of   the   route   table
            request . set_RouteT  ableId ( params [' route_tabl
e_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_r  oute_entry ( self ,  route_tabl  e_id ):
        """
        describe_r  oute_entry :  Query   the   route   entry
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36014 . htm
        """
        try :
            request  =  DescribeRo  uteTablesR  equest .
DescribeRo  uteTablesR  equest ()
            request . set_RouteT  ableId ( route_tabl  e_id )
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_r  oute_entry  _status ( self ,  route_tabl
e_id ):
        """
        describe_r  oute_entry  _status :  Query   the   status
of   the   route   entry
        """
        response  =  self . describe_r  oute_entry ( route_tabl
e_id )
        return   response [" RouteTable  s "][" RouteTable "][ 0 ]
[" RouteEntry  s "][" RouteEntry "][ 0 ][" Status "]

def   main ():
    client  =  ACS_CLIENT
    vswitch  =  VSwitch ( client )
    route_tabl  e  =  RouteTable ( client )
    route_entr  y  =  RouteEntry ( client )
```

```
    params  = {}
    params [' route_tabl  e_name '] = " sdk_route_  table "
    params [' destinatio  n_cidr_blo  ck '] = " 0 . 0 . 0 . 0 / 0 "
    params [' nexthop_id '] = " i - bp1e82xlho  b2xxxxxxxx "
    params [' nexthop_ty  pe '] = " Instance "

    params [' vpc_id '] = " vpc - bp15opprpg  0rgxxxxxxx  x "
    params [' vswitch_id '] = " vsw - bp1e67w26n  2sjxxxxxxx  x "

   # Create   a   route   table
    route_tabl  e_json  =  route_tabl  e . create_rou  te_table (
params )
    CommonUtil . log (" create_rou  te_table ",  route_tabl  e_json
)

   # Query   the   vswitch
    vswitch_js  on  =  vswitch . describe_v  switch_att  ribute (
params )
    CommonUtil . log (" describe_v  switch_att  ribute ",
vswitch_js  on )

   # Associate   the   route   table   with   the   vswitch
    params [' route_tabl  e_id '] =  route_tabl  e_json ['
RouteTable  Id ']
    associate_  json  =  route_tabl  e . associate_  route_tabl  e
( params )
    CommonUtil . log (" associate_  route_tabl  e ",  associate_
json )

   # Create   a   route   entry
    create_rou  te_entry_j  son  =  route_entr  y . create_rou
te_entry ( params )
    CommonUtil . log (" create_rou  te_entry ",  create_rou
te_entry_j  son )

   # Delete   the   route   entry
    delete_rou  te_entry_j  son  =  route_entr  y . delete_rou
te_entry ( params )
    CommonUtil . log (" delete_rou  te_entry ",  delete_rou
te_entry_j  son )

   # Disassocia  te   the   route   table   from   the   vswitch
    unassociat  e_json  =  route_tabl  e . unassociat  e_route_ta
ble ( params )
    CommonUtil . log (" unassociat  e_route_ta  ble ",  unassociat
e_json )

   # Delete   the   route   table
    delete_rou  te_table_j  son  =  route_tabl  e . delete_rou
te_table ( params )
    CommonUtil . log (" delete_rou  te_table ",  delete_rou
te_table_j  son )


if   __name__  == " __main__ ":
```

```
    sys . exit ( main ())
```

5. **Access the** *vpc_route_ entry . py* **directory and run the following command to create a custom route entry.**

```
python  vpc_route_ entry . py
```

The system displays the following output:

```
--------------------------- create_rou te_table
 ---------------------------
{
  " RouteTable Id ": " vtb - 8vbn7px9zx wr2xxxxxxx x ",
  " RequestId ": " 8B351EE1 - 614F - 44E4 - 93AF - 1CADA4BF02 E8 "
}

--------------------------- describe_v switch_att ribute
 ---------------------------

{
  " Status ": "",
  " NetworkAcl Id ": "",
  " VpcId ": "",
  " Descriptio n ": "",
  " Ipv6CidrBl ock ": "",
  " CreationTi me ": "",
  " CloudResou rces ": {
    " CloudResou rceSetType ": []
  },
  " ZoneId ": "",
  " ResourceGr oupId ": "",
  " VSwitchId ": "",
  " RequestId ": " 5E199415 - BBA3 - 443D - B1EC - 06341FE267 F4 ",
  " VSwitchNam e ": "",
  " CidrBlock ": ""
}

--------------------------- associate_ route_tabl e
 ---------------------------
{
  " RequestId ": " 5F33E444 - 5CCD - 4677 - 91AB - 3E234A9A64 E4 "
}

--------------------------- create_rou te_entry
 ---------------------------
{
  " RequestId ": " D6035ECA - DD81 - 4FAB - B084 - 55BE60FB18 ED "
}

--------------------------- delete_rou te_entry
 ---------------------------
{
  " RequestId ": " 54108FD7 - 8609 - 4111 - 919D - B2983466F4 80 "
}

--------------------------- unassociat e_route_ta ble
 ---------------------------
{
  " RequestId ": " 0F36A76A - 1E54 - 41DC - 852E - 1D970FDE8F 3F "
}
```

```
--------------------------- delete_rou  te_table
 ---------------------------
{
  " RequestId ": " F3151A59 - 4F90 - 4531 - AFDC - B7B7CF70A8  C1 "
}
```

2.4 Elastic IP (EIP)

2.4.1 Attach an EIP to an ECS instance

This topic describes how to use the Alibaba Cloud Python SDK to attach an EIP to an ECS instance.

Prerequisites

Before you attach an EIP to an ECS instance, you need to ensure that:

- **The ECS instance is of the VPC network type.**
- **The ECS instance and the EIP are locate in the same region.**
- **The ECS instance is in the running or stopped status.**
- **The ECS instance is not attached to a public IP address or another EIP.**

Context

The detailed steps used in this tutorial to attach an EIP to an ECS instance are as follows:

1. **Create an EIP in the China (Hangzhou) region.**
2. **Attach the EIP to the ECS instance.**
3. **Query EIPs attached to the ECS instance.**
4. **Modify the peak bandwidth and name of the EIP.**
5. **Query the modified EIP.**
6. **Disassociate an EIP from an ECS instance.**
7. **Release the EIP.**

Procedure

1. **In the downloaded SDK directory, open the** *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* **folder.**
2. **Open the** *consts . py* **file in your text editor and configure the** `ACS_CLIENT` **parameter used for user authentication.**
3. **In the downloaded SDK directory, open the** *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ eip* **folder.**

4. **Open the *eip_quick_ start . py* file in your editor and configure your required parameters. Then, save the configurations and exit the editor.**

```
# encoding = utf - 8
 import   sys
 import   json
 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   AllocateEi
 pAddressRe  quest
 from   aliyunsdkv  pc . request . v20160428   import   AssociateE
 ipAddressR  equest
 from   aliyunsdkv  pc . request . v20160428   import   DescribeEi
 pAddresses  Request
 from   aliyunsdkv  pc . request . v20160428   import   Unassociat
 eEipAddres  sRequest
 from   aliyunsdkv  pc . request . v20160428   import   ModifyEipA
 ddressAttr  ibuteReque  st
 from   aliyunsdkv  pc . request . v20160428   import   ReleaseEip
 AddressReq  uest
 from   sdk_lib . exception   import   ExceptionH  andler
 from   sdk_lib . check_stat  us   import   CheckStatu  s
 from   sdk_lib . consts   import  *
 from   sdk_lib . common_uti  l   import   CommonUtil

"""
 Create   an   EIP -> Attach   the   EIP   to   an   ECS   instance
 -> Query   EIPs -> Modify   the   configurat  ions   and   name
 of   the   EIP -> Query   the   EIP -> Disassocia  te   the   EIP -
> Release   the   EIP
"""
 class   Eip ( object ):
     def   __init__ ( self ,  client ):
         self . client  =  client

     def   allocate_e  ip_address ( self ,  params ):
        """
         allocate_e  ip_address :  Apply   for   an   EIP
         API   reference   link :  https :// www . alibabaclo  ud .
 com / help / doc - detail / 36016 . htm
        """
         try :
             request  =  AllocateEi  pAddressRe  quest . AllocateEi
 pAddressRe  quest ()
             response  =  self . client . do_action_  with_excep
 tion ( request )
             response_j  son  =  json . loads ( response )
             if   CheckStatu  s . check_stat  us ( TIME_DEFAU
 LT_OUT ,  DEFAULT_TI  ME ,
                                       self . describe_e
 ip_status ,
                                       AVAILABLE ,  response_j
 son [" Allocation  Id "]):
                 return   response_j  son
         except   ServerExce  ption   as   e :
             ExceptionH  andler . server_exc  eption ( e )
         except   ClientExce  ption   as   e :
             ExceptionH  andler . client_exc  eption ( e )

     def   associate_  eip_addres  s ( self ,  params ):
        """
```

```
        associate_ eip_addres s : Attach the EIP to a
cloud product instance in the same region
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36017 . htm
       """
        try :
            request = AssociateE ipAddressR equest .
AssociateE ipAddressR equest ()
           # The ID of the EIP
            request . set_Alloca tionId ( params [' allocation
_id '])
           # The type of the cloud product instance
to attach
            request . set_Instan ceType ( params [' instance_t
ype '])
           # The ID of the instance to attach
            request . set_Instan ceId ( params [' instance_i d
'])
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
            if CheckStatu s . check_stat us ( TIME_DEFAU
LT_OUT , DEFAULT_TI ME ,
                                        self . describe_e
ip_status ,
                                        InUse , params ['
allocation _id ']):
                return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_e ip_address ( self , allocation _id ):
       """
        describe_e ip_status : Query created EIPs in a
region .
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36018 . htm
       """
        try :
            request = DescribeEi pAddresses Request .
DescribeEi pAddresses Request ()
           # The ID of the EIP
            request . set_Alloca tionId ( allocation _id )
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_e ip_status ( self , allocation _id ):
       """
        describe_e ip_status : Query the status of
created EIPs in a region .
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36018 . htm
       """
       # The ID of the EIP
        response = self . describe_e ip_address ( allocation
_id )
```

```
        return   response [" EipAddress  es "][" EipAddress "][ 0 ]
[" Status "]


    def   unassociat  e_eip_addr  ess ( self ,  params ):
       """
        unassociat  e_eip_addr  ess :  Disassocia  te   an   EIP
from   a   cloud   resource .
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36021 . htm
       """
        try :
            request  =  Unassociat  eEipAddres  sRequest .
Unassociat  eEipAddres  sRequest ()
           #  The   ID   of   the   EIP
            request . set_Alloca  tionId ( params [' allocation
_id '])
           #  The   type   of   the   resource   to   disassocia
te
            request . set_Instan  ceType ( params [' instance_t
ype '])
           #  The   ID   of   the   cloud   product   instance   to
  disassocia  te
            request . set_Instan  ceId ( params [' instance_i  d
'])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                       self . describe_e
ip_status ,
                                       AVAILABLE ,  params ['
allocation  _id ']):
                return   response_j  son
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   modify_eip  _address ( self ,  params ):
       """
        modify_eip  _address :  Modify   the   name ,  descriptio
n ,  and   peak   bandwidth   of   an   EIP
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36019 . htm
       """
        try :
            request  =  ModifyEipA  ddressAttr  ibuteReque  st .
ModifyEipA  ddressAttr  ibuteReque  st ()
           #  The   ID   of   the   EIP
            request . set_Alloca  tionId ( params [' allocation
_id '])
           #  The   peak   bandwidth   in   Mbps   of   the   EIP
            request . set_Bandwi  dth ( params [' bandwidth '])
           #  The   name   of   the   EIP
            request . set_Name ( params [' name '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
```

```
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   release_ei  p_address ( self ,  params ):
       """
        release_ei  p_address :  Release   an   EIP
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36020 . htm
       """
        try :
            request  =  ReleaseEip  AddressReq  uest . ReleaseEip
AddressReq  uest ()
           #  The   ID   of   the   EIP   to   release
            request . set_Alloca  tionId ( params [' allocation
_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

def   main ():

    client  =  ACS_CLIENT
    eip  =  Eip ( client )

    params  = {}

   #  Create   an   EIP
    eip_respon  se_json  =  eip . allocate_e  ip_address ( params )
    CommonUtil . log (" allocate_e  ip_address ",  eip_respon
se_json )

   #  Attach   the   EIP   to   an   ECS   instance
    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]
    params [' instance_i  d '] =  ECS_INSTAN  CE_ID
    params [' instance_t  ype '] = ' EcsInstanc  e '
    eip_respon  se_json  =  eip . associate_  eip_addres  s (
params )
    CommonUtil . log (" associate_  eip_addres  s ",  eip_respon
se_json )

   #  Query   EIPs
    eip_respon  se_json  =  eip . describe_e  ip_address ( params
[' allocation  _id '])
    CommonUtil . log (" describe_e  ip_address ",  eip_respon
se_json )

   #  Modify   the   configurat  ions   and   name   of   an   EIP
    params [' bandwidth '] =  BANDWIDTH_  50
    params [' name '] =  EIP_NEW_NA  ME
    eip_respon  se_json  =  eip . modify_eip  _address ( params )
    CommonUtil . log (" modify_eip  _address ",  eip_respon
se_json )

   #  Query   the   EIP
    eip_respon  se_json  =  eip . describe_e  ip_address ( params
[' allocation  _id '])
    CommonUtil . log (" describe_e  ip_address ",  eip_respon
se_json )
```

```
    #  Disassocia  te   the   EIP
     eip_respon  se_json  =  eip . unassociat  e_eip_addr  ess (
params )
     CommonUtil . log (" unassociat  e_eip_addr  ess ",  eip_respon
se_json )

    #  Release   the   EIP
     eip_respon  se_json  =  eip . release_ei  p_address ( params )
     CommonUtil . log (" release_ei  p_address ",  eip_respon
se_json )

if   __name__  == ' __main__ ':
    sys . exit ( main ())
```

5. **Access the** *eip_quick_ start . py* **directory and run the following command to attach the EIP to the ECS instance.**

```
python   eip_quick_  start . py
```

The system displays the following output:

```
--------------------------- allocate_e  ip_address
 ---------------------------
{
  " EipAddress ": " 47 . xx . xx . 23 ",
  " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
  " RequestId ": " C438312E - F7A4 - 4A04 - 901F - D22FE23EDB  4D ",
  " Allocation  Id ": " eip - bp1wybucvh  hx5xxxxxxx  x "
}
--------------------------- associate_  eip_addres  s
 ---------------------------
{
  " RequestId ": " 6EC6605E - 3D2B - 4EE8 - BD13 - F1964CD1EA  B1 "
}
--------------------------- describe_e  ip_address
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " PageSize ":  10 ,
  " EipAddress  es ": {
    " EipAddress ": [
      {
        " ISP ": " BGP ",
        " ExpiredTim  e ": "",
        " InternetCh  argeType ": " PayByBandw  idth ",
        " IpAddress ": " 47 . xx . xx . 23 ",
        " Allocation  Id ": " eip - bp1wybucvh  hx5xxxxxxx  x ",
        " PrivateIpA  ddress ": "",
        " Status ": " InUse ",
        " BandwidthP  ackageId ": "",
        " InstanceId ": " i - bp1e82xlho  b2xxxxxxxx ",
        " InstanceRe  gionId ": " cn - hangzhou ",
        " RegionId ": " cn - hangzhou ",
        " AvailableR  egions ": {
          " AvailableR  egion ": [
            " cn - hangzhou "
          ]
        },
        " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
```

```
        " HasReserva  tionData ":  false ,
        " InstanceTy  pe ": " EcsInstanc  e ",
        " Allocation  Time ": " 2019 - 04 - 17T11 : 57 : 43Z ",
        " Name ": "",
        " OperationL  ocks ": {
          " LockReason ": []
        },
        " Mode ": " NAT ",
        " BandwidthP  ackageType ": "",
        " BandwidthP  ackageBand  width ": "",
        " Bandwidth ": " 5 ",
        " HDMonitorS  tatus ": " OFF ",
        " ChargeType ": " PostPaid ",
        " SecondLimi  ted ":  false ,
        " Descritpio  n ": ""
      }
    ]
  },
  " RequestId ": " 8715A878 - A808 - 4CC4 - AAD5 - E414FDAB5B  0E "
}
--------------------------- modify_eip  _address
 ---------------------------
{
  " RequestId ": " 2108AE1C - 94FB - 475D - BFEE - EC88598BF6  A6 "
}
--------------------------- describe_e  ip_address
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " PageSize ":  10 ,
  " EipAddress  es ": {
    " EipAddress ": [
      {
        " ISP ": " BGP ",
        " ExpiredTim  e ": "",
        " InternetCh  argeType ": " PayByBandw  idth ",
        " IpAddress ": " 47 . xx . xx . 23 ",
        " Allocation  Id ": " eip - bp1wybucvh  hx5xxxxxx  x ",
        " PrivateIpA  ddress ": "",
        " Status ": " InUse ",
        " BandwidthP  ackageId ": "",
        " InstanceId ": " i - bp1e82xlho  b2xxxxxxx ",
        " InstanceRe  gionId ": " cn - hangzhou ",
        " RegionId ": " cn - hangzhou ",
        " AvailableR  egions ": {
          " AvailableR  egion ": [
            " cn - hangzhou "
          ]
        },
        " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
        " HasReserva  tionData ":  false ,
        " InstanceTy  pe ": " EcsInstanc  e ",
        " Allocation  Time ": " 2019 - 04 - 17T11 : 57 : 43Z ",
        " Name ": " EIP_NEW_NA  ME ",
        " OperationL  ocks ": {
          " LockReason ": []
        },
        " Mode ": " NAT ",
        " BandwidthP  ackageType ": "",
        " BandwidthP  ackageBand  width ": "",
        " Bandwidth ": " 50 ",
        " HDMonitorS  tatus ": " OFF ",
        " ChargeType ": " PostPaid ",
```

```
        " SecondLimi  ted ":  false ,
        " Descritpio  n ": ""
      }
    ]
  },
  " RequestId ": " 6694D35B - B5DD - 4506 - 8AB1 - 2D16477646  DE "
}
--------------------------- unassociat  e_eip_addr  ess
 ---------------------------
{
  " RequestId ": " EDE86CF6 - EE68 - 4922 - B919 - 85A4F11BF6  68 "
}
--------------------------- release_ei  p_address
 ---------------------------
{
  " RequestId ": " 53FEE062 - B595 - 4D64 - AB47 - 834015D328  88 "
}
```

2.4.2 Add an EIP to an Internet Shared Bandwidth

This topic describes how to use the Alibaba Cloud Python SDK to add an EIP to an Internet Shared Bandwidth.

Context

The detailed steps used in this tutorial to add an EIP to an Internet Shared Bandwidth are as follows:

1. **Create an EIP in the China (Hangzhou) region.**
2. **Create an Internet Shared Bandwidth in the China (Hangzhou) region.**
3. **Add the EIP to the Internet Shared Bandwidth.**
4. **Query the Internet Shared Bandwidth.**
5. **Remove the EIP from the Internet Shared Bandwidth.**
6. **Delete the Internet Shared Bandwidth.**
7. **Release the EIP.**

Procedure

1. **In the downloaded SDK directory, open the** *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* **folder.**
2. **Open the** *consts . py* **file in your text editor and configure the** `ACS_CLIENT` **parameter used for authentication.**
3. **In the downloaded SDK directory, open the** *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ eip* **folder.**

4. **Open the *eip_add_cb wp . py* file in your editor and configure your required parameters. Then, save the configurations and exit the editor.**

```
# encoding = utf - 8
 import   sys
 from   aliyunsdkv  pc . request . v20160428   import   CreateComm
 onBandwidt  hPackageRe  quest
 from   aliyunsdkv  pc . request . v20160428   import   AddCommonB
 andwidthPa  ckageIpReq  uest
 from   aliyunsdkv  pc . request . v20160428   import   DescribeCo
 mmonBandwi  dthPackage  sRequest
 from   aliyunsdkv  pc . request . v20160428   import   RemoveComm
 onBandwidt  hPackageIp  Request
 from   aliyunsdkv  pc . request . v20160428   import   DeleteComm
 onBandwidt  hPackageRe  quest
 from   sdk_lib . common_uti  l   import   CommonUtil
 from   sdk_lib . sdk_eip   import  *

"""
 Create   an   EIP -> Create   an   Internet   Shared   Bandwidth ->
 The   ID   of   the   Internet   Shared   Bandwidth -> Add   the
 EIP
 to   the   Internet   Shared   Bandwidth -> Query   the   Internet
   Shared   Bandwidth -> Remove   the   EIP   from   the
 Internet   Shared   Bandwidth -> Delete   the   Internet   Shared
 Bandwidth -> Release   the   EIP
"""
 class   CommonBand  widthPacka  ge ( object ):
     def   __init__ ( self ,  client ):
         self . client  =  client

     def   create_com  mon_bandwi  dth_packag  e ( self ,  params ):
        """
         create_com  mon_bandwi  dth_packag  e :  Create   an
 Internet   Shared   Bandwidth
         API   reference   link :  https :// www . alibabaclo  ud .
 com / help / doc - detail / 55930 . htm
        """
         try :
             request  =  CreateComm  onBandwidt  hPackageRe  quest .
 CreateComm  onBandwidt  hPackageRe  quest ()
            #  The   peak   bandwidth   in   Mbps   of   the
 Internet   Shared   Bandwidth
             request . set_Bandwi  dth ( params [' bandwidth '])
             response  =  self . client . do_action_  with_excep
 tion ( request )
             response_j  son  =  json . loads ( response )
             if   CheckStatu  s . check_stat  us ( TIME_DEFAU
 LT_OUT ,  DEFAULT_TI  ME ,
                                        self . describe_c
 bwp_status ,
                                        AVAILABLE ,  response_j
 son [" BandwidthP  ackageId "]):
                 return   response_j  son
         except   ServerExce  ption   as   e :
             ExceptionH  andler . server_exc  eption ( e )
         except   ClientExce  ption   as   e :
             ExceptionH  andler . client_exc  eption ( e )

     def   add_common  _bandwidth  _packageIp ( self ,  params ):
        """
```

```
        add_common _bandwidth _packageIp : Add the EIP to
  the Internet Shared Bandwidth
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 55989 . htm
       """
        try :
            request = AddCommonB andwidthPa ckageIpReq uest .
AddCommonB andwidthPa ckageIpReq uest ()
           # The ID of the EIP instance
            request . set_IpInst anceId ( params [' ip_instanc
e_id ']) 
           # The ID of the Internet Shared Bandwidth
            request . set_Bandwi dthPackage Id ( params ['
bandwidth_ package_id ']) 
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_c bwp ( self , cbwp_id ):
       """
        describe_c bwp : Query the Internet Shared
Bandwidth
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 55997 . htm
       """
        try :
            request = DescribeCo mmonBandwi dthPackage
sRequest . DescribeCo mmonBandwi dthPackage sRequest ()
           # The ID of the Internet Shared Bandwidth
            request . set_Bandwi dthPackage Id ( cbwp_id )
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_c bwp_status ( self , cbwp_id ):
       """
        describe_c bwp : Query the status of the
Internet Shared Bandwidth
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 55997 . htm
       """
       # The ID of the Internet Shared Bandwidth
        response = self . describe_c bwp ( cbwp_id )
        return response [" CommonBand widthPacka ges "]["
CommonBand widthPacka ge "][ 0 ][" Status "]


    def remove_com mon_bandwi dth_packag eIp ( self , params
):
       """
        remove_com mon_bandwi dth_packag eIp : Remove the
EIP from the Internet Shared Bandwidth
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 55995 . htm
```

```
        """
        try :
            request = RemoveComm onBandwidt hPackageIp Request
. RemoveComm onBandwidt hPackageIp Request ()
            # The ID of the EIP
            request . set_IpInst anceId ( params [' ip_instanc
e_id '])
            # The ID of the Internet Shared Bandwidth
            request . set_Bandwi dthPackage Id ( params ['
bandwidth_ package_id '])
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )


    def delete_com mon_bandwi dth_packag e ( self , params ):
        """
        delete_com mon_bandwi dth_packag e : Delete the
Internet Shared Bandwidth
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 56000 . htm
        """
        try :
            request = DeleteComm onBandwidt hPackageRe quest .
DeleteComm onBandwidt hPackageRe quest ()
            # The ID of the Internet Shared Bandwidth
            request . set_Bandwi dthPackage Id ( params ['
bandwidth_ package_id '])
            # Indicate whether to forcibly delete the
Internet Shared Bandwidth
            request . set_Force ( params [' force '])
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

def main ():

    client = ACS_CLIENT
    eip = Eip ( client )
    cbwp = CommonBand widthPacka ge ( client )

    params = {}

   # Create an EIP
    eip_respon se_json = eip . allocate_e ip_address ( params )
    CommonUtil . log (" allocate_e ip_address ", eip_respon
se_json )
    params [' allocation _id '] = eip_respon se_json ["
Allocation Id "]

   # Create an Internet Shared Bandwidth
    params [' allocation _id '] = eip_respon se_json ["
Allocation Id "]
```

```
    params [' ip_instanc  e_id '] =  eip_respon  se_json ["
Allocation  Id "]
    params [' bandwidth '] =  BANDWIDTH_  10
    cbwp_repso  nse_json  =  cbwp . create_com  mon_bandwi
dth_packag  e ( params )
    CommonUtil . log (" create_com  mon_bandwi  dth_packag  e ",
cbwp_repso  nse_json )

   #  Add   the   EIP   to   the   Internet   Shared   Bandwidth
    params [' bandwidth_  package_id '] =  cbwp_repso  nse_json ['
BandwidthP  ackageId ']
    cbwp_repso  nse_json  =  cbwp . add_common  _bandwidth
_packageIp ( params )
    CommonUtil . log (" add_common  _bandwidth  _packageIp ",
cbwp_repso  nse_json )

   #  Query   the   Internet   Shared   Bandwidth
    cbwp_repso  nse_json  =  cbwp . describe_c  bwp ( params ['
bandwidth_  package_id '])
    CommonUtil . log (" add_common  _bandwidth  _packageIp ",
cbwp_repso  nse_json )

   #  Remove   the   EIP   from   the   Internet   Shared
Bandwidth
    cbwp_repso  nse_json  =  cbwp . remove_com  mon_bandwi
dth_packag  eIp ( params )
    CommonUtil . log (" remove_com  mon_bandwi  dth_packag  eIp ",
cbwp_repso  nse_json )

   #  Delete   the   Internet   Shared   Bandwidth
    params [' force '] =  True
    cbwp_repso  nse_json  =  cbwp . delete_com  mon_bandwi
dth_packag  e ( params )
    CommonUtil . log (" delete_com  mon_bandwi  dth_packag  e ",
cbwp_repso  nse_json )

   #  Release   the   EIP
    eip_respon  se_json  =  eip . release_ei  p_address ( params )
    CommonUtil . log (" release_ei  p_address ",  eip_respon
se_json )

if   __name__  == ' __main__ ':
    sys . exit ( main ())
```

5. **Access the** *eip_add_cb wp . py* **directory and run the following command to add the EIP to the Internet Shared Bandwidth.**

```
python   eip_add_cb  wp . py
```

The system displays the following output:

```
--------------------------- allocate_e  ip_address
 ---------------------------
{
  " EipAddress ": " 118 . xx . xx . 198 ",
  " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
  " RequestId ": " A830A607 - B7C4 - 49FE - A6EE - 7237D64CDE  2D ",
  " Allocation  Id ": " eip - bp1mdyvr22  qvgxxxxxx  x "
}
--------------------------- create_com  mon_bandwi  dth_packag  e
 ----------------------
```

```
-----
{
  " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
  " BandwidthP  ackageId ": " cbwp - bp12k058pj  jiexxxxxxx  x ",
  " RequestId ": " 93127320 - DD79 - 4F83 - A3B9 - DC99D0597B  0C "
}
--------------------------- add_common  _bandwidth  _packageIp
 ----------------------
----
{
  " RequestId ": " 7F314AFE - B398 - 4348 - AF61 - B7D27B7312  86 "
}
--------------------------- add_common  _bandwidth  _packageIp
 ----------------------
----
{
  " TotalCount ":  1 ,
  " CommonBand  widthPacka  ges ": {
    " CommonBand  widthPacka  ge ": [
      {
        " Status ": " Available ",
        " PublicIpAd  dresses ": {
          " PublicIpAd  dresse ": [
            {
              " IpAddress ": " 118 . xx . xx . 198 ",
              " Allocation  Id ": " eip - bp1mdyvr22  qvgxxxxxxx  x
 "
            }
          ]
        },
        " BusinessSt  atus ": " Normal ",
        " RegionId ": " cn - hangzhou ",
        " BandwidthP  ackageId ": " cbwp - bp12k058pj  jiexxxxxxx  x
 ",
        " Name ": "",
        " ISP ": " BGP ",
        " CreationTi  me ": " 2019 - 04 - 18T01 : 46 : 17Z ",
        " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
        " Bandwidth ": " 10 ",
        " InstanceCh  argeType ": " PostPaid ",
        " HasReserva  tionData ":  false ,
        " InternetCh  argeType ": " PayByBandw  idth ",
        " ExpiredTim  e ": "",
        " Ratio ":  100 ,
        " Descriptio  n ": ""
      }
    ]
  },
  " PageNumber ":  1 ,
  " RequestId ": " 015DD0FA - 742B - 4431 - 92EA - E3F03FDEB8  CD ",
  " PageSize ":  10
}
--------------------------- remove_com  mon_bandwi  dth_packag  eIp
 --------------------
-------
{
  " RequestId ": " A49C9126 - B703 - 4D34 - B552 - A7FE283FB5  DD "
}
--------------------------- delete_com  mon_bandwi  dth_packag  e
 ----------------------
-----
{
  " RequestId ": " E423F648 - C169 - 4B63 - A2CF - 5E6C8E441D  E1 "
}
```

```
--------------------------- release_ei  p_address
 ---------------------------
{
  " RequestId ": " 7E0D34AE - 58C3 - 468A - B021 - 378F8938AE  6B "
}
```

2.4.3 Modify the peak bandwidth of an EIP

This topic describes how to use the Alibaba Cloud Python SDK to modify the peak bandwidth of an EIP.

Context

The example used in the procedure of this topic includes the following steps:

1. **Create an EIP in the China (Hangzhou) region.**
2. **Modify the peak bandwidth of the EIP to 50 Mbit/s.**
3. **Query the modified EIP.**
4. **Modify the peak bandwidth of the EIP to 10 Mbit/s.**
5. **Query the modified EIP.**
6. **Release the EIP.**

Procedure

1. **In the SDK directory, open the** *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* **folder.**
2. **Open the** *consts . py* **file in your text editor and set the** `ACS_CLIENT` **parameter for user authentication.**
3. **In the SDK directory, open the** *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ eip* **folder.**
4. **Open the** *eip_modify _attribute . py* **file in your text editor, set the parameters as needed, and then save the settings and exit the editor.**

```
# encoding = utf - 8
 import   sys
 import   json
 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   AllocateEi
 pAddressRe  quest
 from   aliyunsdkv  pc . request . v20160428   import   AssociateE
 ipAddressR  equest
 from   aliyunsdkv  pc . request . v20160428   import   DescribeEi
 pAddresses  Request
 from   aliyunsdkv  pc . request . v20160428   import   Unassociat
 eEipAddres  sRequest
 from   aliyunsdkv  pc . request . v20160428   import   ModifyEipA
 ddressAttr  ibuteReque  st
```

```
from aliyunsdkvpc.request.v20160428 import ReleaseEipAddressRequest
from sdk_lib.exception import ExceptionHandler
from sdk_lib.check_status import CheckStatus
from sdk_lib.consts import *
from sdk_lib.common_util import CommonUtil

"""
Create an EIP -> Modify the peak bandwidth of the EIP to 50 Mbit/s -> Query the EIP -> Modify the peak bandwidth of the EIP to 10 Mbit/s -> Query the EIP -> Release the EIP
"""
class Eip(object):
    def __init__(self, client):
        self.client = client

    def allocate_eip_address(self, params):
        """
        allocate_eip_address: Create an EIP.
        API reference link: https://www.alibabacloud.com/help/doc-detail/36016.htm
        """
        try:
            request = AllocateEipAddressRequest.AllocateEipAddressRequest()
            response = self.client.do_action_with_exception(request)
            response_json = json.loads(response)
            if CheckStatus.check_status(TIME_DEFAULT_OUT, DEFAULT_TIME,
                                        self.describe_eip_status,
                                        AVAILABLE, response_json["AllocationId"]):
                return response_json
        except ServerException as e:
            ExceptionHandler.server_exception(e)
        except ClientException as e:
            ExceptionHandler.client_exception(e)

    def associate_eip_address(self, params):
        """
        associate_eip_address: Associate the EIP with an instance in the same region.
        API reference link: https://www.alibabacloud.com/help/doc-detail/36017.htm
        """
        try:
            request = AssociateEipAddressRequest.AssociateEipAddressRequest()
            # The ID of the EIP
            request.set_AllocationId(params['allocation_id'])
            # The type of the instance with which the EIP is associated
            request.set_InstanceType(params['instance_type'])
            # The ID of the instance with which the EIP is associated
            request.set_InstanceId(params['instance_id'])
            response = self.client.do_action_with_exception(request)
```

```
            response_j  son  =  json . loads ( response )
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                        self . describe_e
ip_status ,
                                        InUse ,  params ['
allocation  _id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_e  ip_address ( self ,  allocation  _id ):
        """
        describe_e  ip_status :  Query   the   EIP   created   in
the   specified   region .
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36018 . htm
        """
        try :
            request  =  DescribeEi  pAddresses  Request .
DescribeEi  pAddresses  Request ()
           #  The   ID   of   the   EIP
            request . set_Alloca  tionId ( allocation  _id )
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_e  ip_status ( self ,  allocation  _id ):
        """
        describe_e  ip_status :  Query   the   status   of   the
EIP   created   in   the   specified   region .
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36018 . htm
        """
       #  The   ID   of   the   EIP
        response  =  self . describe_e  ip_address ( allocation
_id )
        return   response [" EipAddress  es "][" EipAddress "][ 0 ]
[" Status "]


    def   unassociat  e_eip_addr  ess ( self ,  params ):
        """
        unassociat  e_eip_addr  ess :  Disassocia  te   the   EIP
from   the   instance .
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36021 . htm
        """
        try :
            request  =  Unassociat  eEipAddres  sRequest .
Unassociat  eEipAddres  sRequest ()
           #  The   ID   of   the   EIP
            request . set_Alloca  tionId ( params [' allocation
_id '])
           #  The   type   of   the   instance   from   which   the
  EIP   is   disassocia  ted
```

```
            request . set_Instan  ceType ( params [' instance_t
ype '])
            #  The   ID   of   the   instance   from   which   the
EIP   is   disassocia  ted
            request . set_Instan  ceId ( params [' instance_i  d
'])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                        self . describe_e
ip_status ,
                                        AVAILABLE ,  params ['
allocation  _id ']):
                return   response_j  son
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   modify_eip  _address ( self ,  params ):
        """
        modify_eip  _address :  Modify   the   name ,  descriptio
n ,  and   peak   bandwidth   of   the   specified   EIP .
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36019 . htm
        """
        try :
            request  =  ModifyEipA  ddressAttr  ibuteReque  st .
ModifyEipA  ddressAttr  ibuteReque  st ()
            #  The   ID   of   the   EIP
            request . set_Alloca  tionId ( params [' allocation
_id '])
            #  The   peak   bandwidth  ( Mbit / s )  of   the   EIP
            request . set_Bandwi  dth ( params [' bandwidth '])
            #  The   name   of   the   EIP
            request . set_Name ( params [' name '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   release_ei  p_address ( self ,  params ):
        """
        release_ei  p_address :  Release   the   specified   EIP .
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36020 . htm
        """
        try :
            request  =  ReleaseEip  AddressReq  uest . ReleaseEip
AddressReq  uest ()
            #  The   ID   of   the   EIP   to   be   released
            request . set_Alloca  tionId ( params [' allocation
_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
```

```
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

def   main ():

    client  =  ACS_CLIENT
    eip  =  Eip ( client )

    params  = {}

   #  Create   an   EIP .
    eip_respon  se_json  =  eip . allocate_e  ip_address ( params )
    CommonUtil . log (" allocate_e  ip_address ",  eip_respon
se_json )

    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]

   #  Modify   the   peak   bandwidth   of   the   EIP   to   50
Mbit / s .
    params [' name '] =  EIP_NEW_NA  ME
    params [' bandwidth '] =  BANDWIDTH_  50
    eip_respon  se_json  =  eip . modify_eip  _address ( params )
    CommonUtil . log (" modify_eip  _address ",  eip_respon
se_json )

   #  Query   the   EIP .
    eip_respon  se_json  =  eip . describe_e  ip_address ( params
[' allocation  _id '])
    CommonUtil . log (" describe_e  ip_address ",  eip_respon
se_json )

   #  Modify   the   peak   bandwidth   of   the   EIP   to   10
Mbit / s .
    params [' bandwidth '] =  BANDWIDTH_  10
    eip_respon  se_json  =  eip . modify_eip  _address ( params )
    CommonUtil . log (" modify_eip  _address ",  eip_respon
se_json )

   #  Query   the   EIP .
    eip_respon  se_json  =  eip . describe_e  ip_address ( params
[' allocation  _id '])
    CommonUtil . log (" describe_e  ip_address ",  eip_respon
se_json )

   #  Release   the   EIP .
    eip_respon  se_json  =  eip . release_ei  p_address ( params )
    CommonUtil . log (" release_ei  p_address ",  eip_respon
se_json )

if   __name__  == ' __main__ ':
```

```
    sys . exit ( main ())
```

5. Enter the directory where the *eip_modify _attribute . py* file is located, and then run the following command to modify the peak bandwidth of the EIP.

```
python  eip_modify  _attribute . py
```

The system output is displayed as follows:

```
--------------------------- allocate_e  ip_address
 ---------------------------
{
  " EipAddress ": " 47 . xx . xx . 225 ",
  " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
  " RequestId ": " 9318DD7A - F065 - 4EA6 - 9EA0 - 20A9C46EDA  DC ",
  " Allocation  Id ": " eip - bp15bzjk5d  jcsxxxxxxx  x "
}
--------------------------- modify_eip  _address
 ---------------------------
{
  " RequestId ": " C39D55A1 - 6B47 - 489B - 8614 - FDB9736EDE  73 "
}
--------------------------- describe_e  ip_address
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " PageSize ":  10 ,
  " EipAddress  es ": {
    " EipAddress ": [
      {
        " ISP ": " BGP ",
        " ExpiredTim  e ": "",
        " InternetCh  argeType ": " PayByBandw  idth ",
        " IpAddress ": " 47 . xx . xx . 225 ",
        " Allocation  Id ": " eip - bp15bzjk5d  jcsxxxxxxx  x ",
        " PrivateIpA  ddress ": "",
        " Status ": " Available ",
        " BandwidthP  ackageId ": "",
        " InstanceId ": "",
        " InstanceRe  gionId ": "",
        " RegionId ": " cn - hangzhou ",
        " AvailableR  egions ": {
          " AvailableR  egion ": [
            " cn - hangzhou "
          ]
        },
        " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
        " HasReserva  tionData ":  false ,
        " InstanceTy  pe ": "",
        " Allocation  Time ": " 2019 - 04 - 18T04 : 01 : 28Z ",
        " Name ": " EIP_NEW_NA  ME ",
        " OperationL  ocks ": {
          " LockReason ": []
        },
        " Mode ": " NAT ",
        " BandwidthP  ackageType ": "",
        " BandwidthP  ackageBand  width ": "",
        " Bandwidth ": " 50 ",
        " HDMonitorS  tatus ": " OFF ",
        " ChargeType ": " PostPaid ",
```

```
        " SecondLimi  ted ":  false ,
        " Descritpio  n ": ""
      }
    ]
  },
  " RequestId ": " 51ECAB45 - 2518 - 4A46 - 89DC - 8ADEE1AFDB  E9 "
}
--------------------------- modify_eip  _address
 ---------------------------
{
  " RequestId ": " ACAB5724 - D05C - 46A4 - 8C2B - 6064AEEC79  2B "
}
--------------------------- describe_e  ip_address
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " PageSize ":  10 ,
  " EipAddress  es ": {
    " EipAddress ": [
      {
        " ISP ": " BGP ",
        " ExpiredTim  e ": "",
        " InternetCh  argeType ": " PayByBandw  idth ",
        " IpAddress ": " 47 . xx . xx . 225 ",
        " Allocation  Id ": " eip - bp15bzjk5d  jcsxxxxxxx  x ",
        " PrivateIpA  ddress ": "",
        " Status ": " Available ",
        " BandwidthP  ackageId ": "",
        " InstanceId ": "",
        " InstanceRe  gionId ": "",
        " RegionId ": " cn - hangzhou ",
        " AvailableR  egions ": {
          " AvailableR  egion ": [
            " cn - hangzhou "
          ]
        },
        " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
        " HasReserva  tionData ":  false ,
        " InstanceTy  pe ": "",
        " Allocation  Time ": " 2019 - 04 - 18T04 : 01 : 28Z ",
        " Name ": " EIP_NEW_NA  ME ",
        " OperationL  ocks ": {
          " LockReason ": []
        },
        " Mode ": " NAT ",
        " BandwidthP  ackageType ": "",
        " BandwidthP  ackageBand  width ": "",
        " Bandwidth ": " 10 ",
        " HDMonitorS  tatus ": " OFF ",
        " ChargeType ": " PostPaid ",
        " SecondLimi  ted ":  false ,
        " Descritpio  n ": ""
      }
    ]
  },
  " RequestId ": " 6F653A80 - AB28 - 4842 - 84A8 - CD444EB81A  29 "
}
--------------------------- release_ei  p_address
 ---------------------------
{
  " RequestId ": " C407633A - 5658 - 482F - AB5E - 069028C3B0  6C "
```

```
}
```

2.4.4 Attach an EIP to an SLB instance or an ENI

This topic describes how to use the Alibaba Cloud Python SDK to attach an EIP to an SLB instance or an ENI.

Prerequisites

Before you attach an EIP to an SLB instance or an ENI, you must ensure that a VPC and a VSwitch are created in the China (Hangzhou) region and an ENI is created under the VPC and the VSwitch.

Context

The detailed steps used in this tutorial to attach an EIP to an SLB instance/ENI are as follows:

1. Create an SLB instance in the China (Hangzhou) region.
2. Create an EIP in the China (Hangzhou) region.
3. Attach the EIP to the SLB instance.
4. Detach the EIP from the SLB instance.
5. Attach the detached EIP to the ENI.
6. Detach the EIP from the ENI.
7. Delete the SLB instance.

Procedure

1. In the downloaded SDK directory, open the `$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib` folder.
2. Open the `consts . py` file in your text editor and configure the `ACS_CLIENT` parameter used for authentication.
3. In the downloaded SDK directory, open the `$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ eip` folder.
4. Open the `eip_associ ate_slb . py` file in your editor and configure your required parameters. Then, save the configurations and exit the editor.

```
# encoding = utf - 8
 import   sys
 import   json
 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   AllocateEi
 pAddressRe  quest
```

```
from aliyunsdkvpc.request.v20160428 import AssociateEipAddressRequest
from aliyunsdkvpc.request.v20160428 import DescribeEipAddressesRequest
from aliyunsdkvpc.request.v20160428 import UnassociateEipAddressRequest
from aliyunsdkvpc.request.v20160428 import ModifyEipAddressAttributeRequest
from aliyunsdkvpc.request.v20160428 import ReleaseEipAddressRequest
from sdk_lib.exception import ExceptionHandler
from sdk_lib.check_status import CheckStatus
from sdk_lib.consts import *
from sdk_lib.common_util import CommonUtil
from sdk_lib.sdk_load_balancer import LoadBalancer

"""
Create a VPC -> Create a VSwitch -> Call an ECS API action to generate an ENI ( these steps are not included
in the following codes )
Create an SLB instance -> Create an EIP -> Attach the EIP to the SLB instance -> Detach the EIP from the SLB instance
-> Attach the EIP to the ENI -> Detach the EIP from the ENI -> Delete the SLB instance
"""
class Eip ( object ):
    def __init__ ( self , client ):
        self . client = client

    def allocate_eip_address ( self , params ):
        """
        allocate_eip_address : Create an EIP
        API reference link : https :// www . alibabacloud .
com / help / doc - detail / 36016 . htm
        """
        try :
            request = AllocateEipAddressRequest . AllocateEi
pAddressRequest ()
            response = self . client . do_action_with_excep
tion ( request )
            response_json = json . loads ( response )
            if CheckStatus . check_status ( TIME_DEFAU
LT_OUT , DEFAULT_TIME ,
                                            self . describe_e
ip_status ,
                                            AVAILABLE , response_j
son [" AllocationId "]):
                return response_json
        except ServerException as e :
            ExceptionHandler . server_exception ( e )
        except ClientException as e :
            ExceptionHandler . client_exception ( e )

    def associate_eip_address ( self , params ):
        """
        associate_eip_address : Attach the EIP to an
cloud product instance in the same region
        API reference link : https :// www . alibabacloud .
com / help / doc - detail / 36017 . htm
        """
        try :
```

```
            request = AssociateE ipAddressR equest .
AssociateE ipAddressR equest ()
            # The ID of the EIP
            request . set_Alloca tionId ( params [' allocation
_id '])
            # The type of the cloud product instance
to attach
            request . set_Instan ceType ( params [' instance_t
ype '])
            # The ID of the instance to attach
            request . set_Instan ceId ( params [' instance_i d
'])
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
            if CheckStatu s . check_stat us ( TIME_DEFAU
LT_OUT , DEFAULT_TI ME ,
                                        self . describe_e
ip_status ,
                                        InUse , params ['
allocation _id ']):
                return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_e ip_address ( self , allocation _id ):
        """
        describe_e ip_status : Query one or more EIPs
in a region
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36018 . htm
        """
        try :
            request = DescribeEi pAddresses Request .
DescribeEi pAddresses Request ()
            # The ID of the EIP
            request . set_Alloca tionId ( allocation _id )
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_e ip_status ( self , allocation _id ):
        """
        describe_e ip_status : Query the status of an
EIP
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36018 . htm
        """
        # The ID of the EIP
        response = self . describe_e ip_address ( allocation
_id )
        return response [" EipAddress es "][" EipAddress "][ 0 ]
[" Status "]


    def unassociat e_eip_addr ess ( self , params ):
        """
```

```
        unassociat  e_eip_addr  ess :  Detach   the   EIP   from
the   cloud   resource
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36021 . htm
       """
        try :
            request  =  Unassociat  eEipAddres  sRequest .
Unassociat  eEipAddres  sRequest ()
           #  The   ID   of   the   EIP
            request . set_Alloca  tionId ( params [' allocation
_id '])
           #  The   type   of   the   resource   to   detach
            request . set_Instan  ceType ( params [' instance_t
ype '])
           #  The   ID   of   the   cloud   product   instance   to
  detach
            request . set_Instan  ceId ( params [' instance_i  d
'])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                        self . describe_e
ip_status ,
                                        AVAILABLE ,  params ['
allocation  _id ']):
                return   response_j  son
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   modify_eip  _address ( self ,  params ):
       """
        modify_eip  _address :  Modify   the   name ,  descriptio
n   and   peak   bandwidth   of   an   EIP
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36019 . htm
       """
        try :
            request  =  ModifyEipA  ddressAttr  ibuteReque  st .
ModifyEipA  ddressAttr  ibuteReque  st ()
           #  The   ID   of   the   EIP
            request . set_Alloca  tionId ( params [' allocation
_id '])
           #  The   peak   bandwidth   in   Mbps   of   the   EIP
            request . set_Bandwi  dth ( params [' bandwidth '])
           #  The   name   of   the   EIP
            request . set_Name ( params [' name '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   release_ei  p_address ( self ,  params ):
       """
        release_ei  p_address :  Release   an   EIP
```

```
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36020 . htm
       """
        try :
            request  =  ReleaseEip  AddressReq  uest . ReleaseEip
AddressReq  uest ()
           #  The   ID   of   the   EIP   to   release
            request . set_Alloca  tionId ( params [' allocation
_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )
def   main ():

    client  =  ACS_CLIENT
    eip  =  Eip ( client )
    load_balan  cer  =  LoadBalanc  er ( client )
   # Create   a   VPC ,  create   a   VSwitch ,  and   call   an
ECS   API   action   to   generate   an   ENI
    params  = {}
    params [' vpc_id '] =  VPC_ID
    params [' vswitch_id '] =  VSWITCH_ID

   # Create   an   SLB   instance
    load_balan  cer_repson  se_json  =  load_balan  cer .
create_loa  d_balancer  _private ( params )
    CommonUtil . log (" create_loa  d_balancer ",  load_balan
cer_repson  se_json )

   # Create   an   EIP
    params [' load_balan  cer_id '] =  load_balan  cer_repson
se_json [' LoadBalanc  erId ']
    params [' instance_i  d '] =  load_balan  cer_repson  se_json
[' LoadBalanc  erId ']
    eip_respon  se_json  =  eip . allocate_e  ip_address ( params )
    CommonUtil . log (" allocate_e  ip_address   slb ",
eip_respon  se_json )

   #  Attach   the   EIP   to   the   SLB   instance
    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]
    params [' instance_t  ype '] = ' SlbInstanc  e '
    eip_respon  se_json  =  eip . associate_  eip_addres  s (
params )
    CommonUtil . log (" associate_  eip_addres  s   eip ",
eip_respon  se_json )

   #  Detach   the   EIP   from   the   SLB   instance
    eip_respon  se_json  =  eip . unassociat  e_eip_addr  ess (
params )
    CommonUtil . log (" unassociat  e_eip_addr  ess   slb ",
eip_respon  se_json )

   #  Attach   the   EIP   to   the   ENI
    params [' instance_i  d '] =  ENI_ID
    params [' instance_t  ype '] = ' NetworkInt  erface '
    eip_respon  se_json  =  eip . associate_  eip_addres  s (
params )
```

```
    CommonUtil . log (" associate_  eip_addres  s   eni ",
eip_respon  se_json )

   #  Detach   the   EIP   from   the   ENI
    eip_respon  se_json  =  eip . unassociat  e_eip_addr  ess (
params )
    CommonUtil . log (" unassociat  e_eip_addr  ess   eni ",
eip_respon  se_json )

   #  Delete   the   SLB   instance
    load_balan  cer_repson  se_json  =  load_balan  cer .
delete_loa  d_balancer ( params )
    CommonUtil . log (" delete_loa  d_balancer ",  load_balan
cer_repson  se_json )

if   __name__  == ' __main__ ':
    sys . exit ( main ())
```

5. **Access the** *eip_associ ate_slb . py* **directory and run the following command to attach the EIP to the SLB instance or the ENI.**

```
Python   eip_associ  ate_slb . py
```

The system displays the following output:

```
--------------------------- create_loa  d_balancer
 ---------------------------
{
  " VpcId ": " vpc - bp15opprpg  0rgxxxxxx  x ",
  " AddressIPV  ersion ": " ipv4 ",
  " LoadBalanc  erName ": " auto_named  _slb ",
  " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
  " VSwitchId ": " vsw - bp1e67w26n  2sjxxxxxx  x ",
  " RequestId ": " D3651A96 - 008C - 4B35 - A36E - 54C2902535  C5 ",
  " Address ": " 172 . xx . xx . 146 ",
  " NetworkTyp  e ": " vpc ",
  " LoadBalanc  erId ": " lb - bp15u6kuma  mmdxxxxxx  x "
}
--------------------------- allocate_e  ip_address   slb
 ---------------------------
{
  " EipAddress ": " 47 . xx . xx . 76 ",
  " ResourceGr  oupId ": " rg - acfm4odxxx  xxxxx ",
  " RequestId ": " 15FD58CD - B186 - 4E2C - B4B3 - 74F7121688  32 ",
  " Allocation  Id ": " eip - bp1ofhmmep  6rkxxxxxx  x "
}
--------------------------- associate_  eip_addres  s   eip
 ---------------------------
{
  " RequestId ": " 5EDABF0B - A067 - 474E - 9556 - 0A3AA87096  0A "
}
--------------------------- unassociat  e_eip_addr  ess   slb
 ---------------------------
-
{
  " RequestId ": " 89556510 - D726 - 490A - 9092 - 9BEA0644CC  43 "
}
--------------------------- associate_  eip_addres  s   eni
 ---------------------------
{
  " RequestId ": " FAE87FDD - 232A - 4859 - 803B - F9B57508AE  DC "
```

```
}
--------------------------- unassociat  e_eip_addr  ess   eni
 ---------------------------
-
{
  " RequestId ": " 7DF556E8 - 12BE - 481A - B83D - B3D9E88365  34 "
}
--------------------------- delete_loa  d_balancer
 ---------------------------
{
  " RequestId ": " 2B70C01A - A440 - 40A5 - A98B - 83A687537C  CE "
}
```

2.5 NAT Gateway

2.5.1 Create a NAT Gateway

This topic describes how to use the Alibaba Cloud Python SDK to create a NAT Gateway.

Context

The detailed steps used in this tutorial to create a NAT Gateway are as follows:

1. **Create a VPC in the China (Shanghai) region.**
2. **Create a VSwitch under the VPC.**
3. **Create a NAT Gateway under the VPC.**
4. **Query the NAT Gateway.**
5. **Delete the NAT Gateway.**
6. **Delete the VSwitch.**
7. **Delete the VPC.**

Procedure

1. **In the downloaded SDK directory, open the** *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* **folder.**
2. **Open the** *consts . py* **file in your text editor and configure the** `ACS_CLIENT` **parameter used for user authentication.**
3. **In the downloaded SDK directory, open the** *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ natgw* **folder.**
4. **Open the** *natgw_quic k_start . py* **file in your text editor and configure your required parameters. Then, save the configurations and exit the text editor.**

```
# encoding = utf - 8
 import   sys
 import   json
```

```
import   time

from   aliyunsdkc  ore . acs_except  ion . exceptions   import
ServerExce  ption ,  ClientExce  ption
from   aliyunsdkv  pc . request . v20160428   import   CreateNatG
atewayRequ  est
from   aliyunsdkv  pc . request . v20160428   import   DeleteNatG
atewayRequ  est
from   aliyunsdkv  pc . request . v20160428   import   DescribeNa
tGatewaysR  equest
from   sdk_lib . sdk_vpc   import   Vpc
from   sdk_lib . sdk_vswitc  h   import   VSwitch
from   sdk_lib . common_uti  l   import   CommonUtil
from   sdk_lib . check_stat  us   import   CheckStatu  s
from   sdk_lib . exception   import   ExceptionH  andler
from   sdk_lib . consts   import  *

client  =  ACS_CLIENT


class   NatGateway ( object ):
    def   __init__ ( self ,  client ):
        self . client  =  client

    def   create_nat  _gateway ( self ,  params ):
       """
        create_nat  _gateway :  Create   a   NAT   Gateway
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36048 . htm
       """
        try :
            request  =  CreateNatG  atewayRequ  est . CreateNatG
atewayRequ  est ()
            request . set_VpcId ( params [' vpc_id '])
            response  =  client . do_action_  with_excep  tion (
request )
            response_j  son  =  json . loads ( response )
           #  Check   if   the   NAT   Gateway   is   in   the
available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                        self . describe_n
at_gateway  _status ,
                                        AVAILABLE ,  response_j
son [' NatGateway  Id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_n  at_gateway ( self ,  nat_gatewa  y_id ):
       """
        describe_n  at_gateway :  Query   NAT   Gateways   in   a
region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36054 . htm
       """
        try :
            request  =  DescribeNa  tGatewaysR  equest .
DescribeNa  tGatewaysR  equest ()
            request . set_NatGat  ewayId ( nat_gatewa  y_id )
            response  =  client . do_action_  with_excep  tion (
request )
```

```
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   delete_nat  _gateway ( self ,  params ):
        """
        delete_nat  _gateway :  Delete   a   NAT   Gateway
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36051 . htm
        """
        try :
            request  =  DeleteNatG  atewayRequ  est . DeleteNatG
atewayRequ  est ()
            request . set_NatGat  ewayId ( params [' nat_gatewa
y_id '])
            response  =  client . do_action_  with_excep  tion (
request )
            response_j  son  =  json . loads ( response )
           #  Check   if   the   NAT   Gateway   is   in   the
available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME  *  5 ,
                                        self . describe_n
at_gateway  _status ,
                                       '',  params [' nat_gatewa
y_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_n  at_gateway  _status ( self ,  nat_gatewa
y_id ):
        """
        describe_n  at_gateway  _status :  Query   the   status
of   NAT   Gateways   in   a   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36054 . htm
        """
        response  =  self . describe_n  at_gateway ( nat_gatewa
y_id )
        if   len ( response [" NatGateway  s "][" NatGateway "])
==  0 :
            return  ''
        return   response [" NatGateway  s "][" NatGateway "][ 0 ]
[' Status ']


def   main ():
    vpc  =  Vpc ( client )
    vswitch  =  VSwitch ( client )
    nat_gatewa  y  =  NatGateway ( client )

    params  = {}

   #  Create   a   VPC
    vpc_json  =  vpc . create_vpc ()
    CommonUtil . log (" create_vpc ",  vpc_json )

   #  Create   a   VSwitch
```

```
    params [' vpc_id '] =  vpc_json [' VpcId ']
    params [' zone_id '] = " cn - shanghai - b "
    params [' cidr_block '] = " 172 . 16 . 1 . 0 / 24 "
    vswitch_js  on  =  vswitch . create_vsw  itch ( params )
    CommonUtil . log (" create_vsw  itch ",  vswitch_js  on )

   #  Create   a   NAT   Gateway
    nat_gatewa  y_json  =  nat_gatewa  y . create_nat  _gateway (
params )
    CommonUtil . log (" create_nat  _gateway ",  nat_gatewa  y_json
)

   #  Query   the   NAT   Gateway
    params [' nat_gatewa  y_id '] =  nat_gatewa  y_json ['
NatGateway  Id ']
    nat_gatewa  y_json  =  nat_gatewa  y . describe_n  at_gateway (
params [' nat_gatewa  y_id '])
    CommonUtil . log (" describe_n  at_gateway ",  nat_gatewa
y_json )

   #  Delete   the   NAT   Gateway
    nat_gatewa  y_json  =  nat_gatewa  y . delete_nat  _gateway (
params )
    CommonUtil . log (" delete_nat  _gateway ",  nat_gatewa  y_json
)

   #  Delete   the   VSwitch
    params [' vswitch_id '] =  vswitch_js  on [' VSwitchId ']
    vswitch_js  on  =  vswitch . delete_vsw  itch ( params )
    CommonUtil . log (" delete_vsw  itch ",  vswitch_js  on )

   #  Delete   the   VPC
    vpc_json  =  vpc . delete_vpc ( params )
    CommonUtil . log (" delete_vpc ",  vpc_json )


if   __name__  == " __main__ ":
    sys . exit ( main ())
```

5. **Access the** *natgw_quic k_start . py* **directory and run the following command to create a NAT Gateway.**

```
python   natgw_quic  k_start . py
```

The system displays the following output:

```
---------------------------- create_vpc ----------------------------
{
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RouteTable  Id ": " vtb - uf6wzp25d8  lkbxxxxxxx  x ",
  " VRouterId ": " vrt - uf6di7voec  myqxxxxxxx  x ",
  " VpcId ": " vpc - uf63cqupgh  mk1xxxxxxx  x ",
  " RequestId ": " 97D36E19 - F789 - 424F - A473 - 660D63EF8C  F9 "
}

---------------------------- create_vsw  itch
 ----------------------------
{
  " VSwitchId ": " vsw - uf6fovepnk  4yexxxxxxx  x ",
  " RequestId ": " 18DA0E81 - 34A6 - 4877 - 9771 - E2C4EEEBAD  D1 "
}
```

```
--------------------------- create_nat  _gateway
 ---------------------------
{
  " NatGateway  Id ": " ngw - uf6mfrcmzk  tstxxxxxxx  x ",
  " BandwidthP  ackageIds ": {
    " BandwidthP  ackageId ": []
  },
  " ForwardTab  leIds ": {
    " ForwardTab  leId ": [
      " ftb - uf6411str8  n9sxxxxxxx  x "
    ]
  },
  " RequestId ": " B1F791C8 - 73B1 - 46C5 - 8A20 - 726A615BC6  27 ",
  " SnatTableI  ds ": {
    " SnatTableI  d ": [
      " stb - uf6t4eijvq  3aexxxxxxx  x "
    ]
  }
}

--------------------------- describe_n  at_gateway
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " RequestId ": " 1F9303B1 - 4024 - 4A92 - B67E - FB6BE1DC76  D1 ",
  " PageSize ":  10 ,
  " NatGateway  s ": {
    " NatGateway ": [
      {
        " Status ": " Available ",
        " BandwidthP  ackageIds ": {
          " BandwidthP  ackageId ": []
        },
        " VpcId ": " vpc - uf63cqupgh  mk1xxxxxxx  x ",
        " Descriptio  n ": "",
        " ForwardTab  leIds ": {
          " ForwardTab  leId ": [
            " ftb - uf6411str8  n9sxxxxxxx  x "
          ]
        },
        " IpLists ": {
          " IpList ": []
        },
        " BusinessSt  atus ": " Normal ",
        " RegionId ": " cn - shanghai ",
        " CreationTi  me ": " 2019 - 04 - 24T09 : 09 : 12Z ",
        " NatGateway  Id ": " ngw - uf6mfrcmzk  tstxxxxxxx  x ",
        " SnatTableI  ds ": {
          " SnatTableI  d ": [
            " stb - uf6t4eijvq  3aexxxxxxx  x "
          ]
        },
        " AutoPay ":  false ,
        " InstanceCh  argeType ": " PostPaid ",
        " ExpiredTim  e ": "",
        " Spec ": " Small ",
        " Name ": ""
      }
    ]
  }
}
```

```
--------------------------- delete_nat  _gateway
 ---------------------------
{
  " RequestId ": " A0B71FE4 - 4756 - 4D91 - 899E - 1DFA52D861  5E "
}

--------------------------- delete_vsw  itch
 ---------------------------
{
  " RequestId ": " F224307E - 3DE4 - 4415 - AE19 - DDCF246954  62 "
}

--------------------------- delete_vpc ---------------------------
{
  " RequestId ": " 1BFFCBC3 - 7F83 - 436C - 96E9 - CA4A620072  DA "
}
```

2.5.2 Associate and disassociate an EIP

This topic describes how to use the Alibaba Cloud Python SDK to associate and disassociate an EIP.

Context

The detailed steps used in this tutorial to associate and disassociate an EIP are as follows:

1. **Create a VPC in the China (Shanghai) region.**
2. **Create a VSwitch under the VPC.**
3. **Create a NAT Gateway under the VPC.**
4. **Create an EIP in the China (Shanghai) region.**
5. **Associate the EIP to the NAT Gateway.**
6. **Query the EIP.**
7. **Create an Internet Shared Bandwidth instance in the China (Shanghai) region.**
8. **Add the EIP to the Internet Shared Bandwidth instance.**
9. **Query the created NAT Gateway.**
10. **Disassociate the EIP from the NAT Gateway.**
11. **Remove the EIP from the Internet Shared Bandwidth instance.**
12. **Delete the Internet Shared Bandwidth instance.**
13. **Delete the NAT Gateway.**
14. **Release the EIP.**
15. **Delete the VSwitch.**
16. **Delete the VPC.**

Procedure

1. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* folder.
2. Open the *consts . py* file in your text editor and configure the ACS_CLIENT parameter used for user authentication.
3. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ natgw* folder.
4. Open the *natgw_asso ciate_eip . py* file in your text editor and configure your required parameters. Then, save the configurations and exit the text editor.

```
# encoding = utf - 8
 import   sys
 import   json
 import   time

 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   CreateNatG
 atewayRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DeleteNatG
 atewayRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DescribeNa
 tGatewaysR  equest
 from   sdk_lib . sdk_vpc   import   Vpc
 from   sdk_lib . sdk_vswitc  h   import   VSwitch
 from   sdk_lib . sdk_eip   import   Eip
 from   sdk_lib . sdk_cbwp   import   CommonBand  widthPacka  ge
 from   sdk_lib . common_uti  l   import   CommonUtil
 from   sdk_lib . check_stat  us   import   CheckStatu  s
 from   sdk_lib . exception   import   ExceptionH  andler
 from   sdk_lib . consts   import  *

 client  =  ACS_CLIENT


 class   NatGateway ( object ):
     def   __init__ ( self ,  client ):
         self . client  =  client

     def   create_nat  _gateway ( self ,  params ):
        """
         create_nat  _gateway :  Create   a   NAT   Gateway
         API   reference   link :  https :// www . alibabaclo  ud .
 com / help / doc - detail / 36048 . htm
        """
         try :
             request  =  CreateNatG  atewayRequ  est . CreateNatG
 atewayRequ  est ()
             request . set_VpcId ( params [' vpc_id '])
             response  =  client . do_action_  with_excep  tion (
 request )
             response_j  son  =  json . loads ( response )
            #  Check   if   the   NAT   Gateway   is   in   the
 available   state
             if   CheckStatu  s . check_stat  us ( TIME_DEFAU
 LT_OUT ,  DEFAULT_TI  ME ,
```

```
                                        self . describe_n
at_gateway  _status ,
                                        AVAILABLE ,  response_j
son [' NatGateway  Id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_n  at_gateway ( self ,  nat_gatewa  y_id ):
        """
        describe_n  at_gateway :  Query   NAT   Gateways   in   a
region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36054 . htm
        """
        try :
            request  =  DescribeNa  tGatewaysR  equest .
DescribeNa  tGatewaysR  equest ()
            request . set_NatGat  ewayId ( nat_gatewa  y_id )
            response  =  client . do_action_  with_excep  tion (
request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   delete_nat  _gateway ( self ,  params ):
        """
        delete_nat  _gateway :  Delete   a   NAT   Gateway
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36051 . htm
        """
        try :
            request  =  DeleteNatG  atewayRequ  est . DeleteNatG
atewayRequ  est ()
            request . set_NatGat  ewayId ( params [' nat_gatewa
y_id '])
            response  =  client . do_action_  with_excep  tion (
request )
            response_j  son  =  json . loads ( response )
          #  Check   if   the   NAT   Gateway   is   in   the
available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME  *  5 ,
                                        self . describe_n
at_gateway  _status ,
                                       '',  params [' nat_gatewa
y_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_n  at_gateway  _status ( self ,  nat_gatewa
y_id ):
        """
        describe_n  at_gateway  _status :  Query   the   status
of   NAT   Gateways   in   a   region
```

```
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36054 . htm
       """
        response  =  self . describe_n  at_gateway ( nat_gatewa
y_id )
        if   len ( response [" NatGateway  s "][" NatGateway "])
==  0 :
            return  ''
        return   response [" NatGateway  s "][" NatGateway "][ 0 ]
[' Status ']


def   main ():
    vpc  =  Vpc ( client )
    vswitch  =  VSwitch ( client )
    eip  =  Eip ( client )
    cbwp  =  CommonBand  widthPacka  ge ( client )
    nat_gatewa  y  =  NatGateway ( client )

    params  = {}

   #  Create   a   VPC
    vpc_json  =  vpc . create_vpc ()
    CommonUtil . log (" create_vpc ",  vpc_json )

   #  Create   a   VSwitch
    params [' vpc_id '] =  vpc_json [' VpcId ']
    params [' zone_id '] = " cn - shanghai - d "
    params [' cidr_block '] = " 172 . 16 . 1 . 0 / 24 "
    vswitch_js  on  =  vswitch . create_vsw  itch ( params )
    CommonUtil . log (" create_vsw  itch ",  vswitch_js  on )

   #  Create   a   NAT   Gateway
    nat_gatewa  y_json  =  nat_gatewa  y . create_nat  _gateway (
params )
    CommonUtil . log (" create_nat  _gateway ",  nat_gatewa  y_json
)

   #  Create   an   EIP
    eip_respon  se_json  =  eip . allocate_e  ip_address ( params )
    CommonUtil . log (" allocate_e  ip_address ",  eip_respon
se_json )
    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]

   #  Associate   the   EIP   to   the   NAT   Gateway
    params [' instance_i  d '] =  nat_gatewa  y_json [' NatGateway
Id ']
    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]
    params [' instance_t  ype '] = ' Nat '
    eip_respon  se_json  =  eip . associate_  eip_addres  s (
params )
    CommonUtil . log (" associate_  eip_addres  s   eip ",
eip_respon  se_json )

   #  Query   the   EIP
    eip_respon  se_json  =  eip . describe_e  ip_address ( params
[' allocation  _id '])
    CommonUtil . log (" describe_e  ip_address ",  eip_respon
se_json )

   #  Create   an   Internet   Shared   Bandwidth
    params [' bandwidth '] =  BANDWIDTH_  10
```

```
    cbwp_repso  nse_json  =  cbwp . create_com  mon_bandwi
dth_packag  e ( params )
    CommonUtil . log (" create_com  mon_bandwi  dth_packag  e ",
cbwp_repso  nse_json )

   #  Add   the   EIP   to   the   Internet   Shared   Bandwidth
    params [' ip_instanc  e_id '] =  params [' allocation  _id ']
    params [' bandwidth_  package_id '] =  cbwp_repso  nse_json ['
BandwidthP  ackageId ']
    cbwp_repso  nse_json  =  cbwp . add_common  _bandwidth
_packageIp ( params )
    CommonUtil . log (" add_common  _bandwidth  _packageIp ",
cbwp_repso  nse_json )

   #  Query   the   NAT   Gateway
    params [' nat_gatewa  y_id '] =  nat_gatewa  y_json ['
NatGateway  Id ']
    nat_gatewa  y_json  =  nat_gatewa  y . describe_n  at_gateway (
params [' nat_gatewa  y_id '])
    CommonUtil . log (" describe_n  at_gateway ",  nat_gatewa
y_json )

   #  Disassocia  te   the   EIP
    eip_respon  se_json  =  eip . unassociat  e_eip_addr  ess (
params )
    CommonUtil . log (" unassociat  e_eip_addr  ess   nat ",
eip_respon  se_json )

   #  Remove   the   EIP   from   the   Internet   Shared
Bandwidth
    cbwp_repso  nse_json  =  cbwp . remove_com  mon_bandwi
dth_packag  eIp ( params )
    CommonUtil . log (" remove_com  mon_bandwi  dth_packag  eIp ",
cbwp_repso  nse_json )

   #  Delete   the   Internet   Shared   Bandwidth
    params [' force '] =  True
    cbwp_repso  nse_json  =  cbwp . delete_com  mon_bandwi
dth_packag  e ( params )
    CommonUtil . log (" delete_com  mon_bandwi  dth_packag  e ",
cbwp_repso  nse_json )

   #  Delete   the   NAT   Gateway
    nat_gatewa  y_json  =  nat_gatewa  y . delete_nat  _gateway (
params )
    CommonUtil . log (" delete_nat  _gateway ",  nat_gatewa  y_json
)

   #  Release   the   EIP
    eip_respon  se_json  =  eip . release_ei  p_address ( params )
    CommonUtil . log (" release_ei  p_address ",  eip_respon
se_json )

   #  Delete   the   VSwitch
    params [' vswitch_id '] =  vswitch_js  on [' VSwitchId ']
    vswitch_js  on  =  vswitch . delete_vsw  itch ( params )
    CommonUtil . log (" delete_vsw  itch ",  vswitch_js  on )

   #  Delete   the   VPC
    vpc_json  =  vpc . delete_vpc ( params )
    CommonUtil . log (" delete_vpc ",  vpc_json )


if   __name__  == " __main__ ":
```

```
    sys . exit ( main ())
```

5. Access the *natgw_asso ciate_eip . py* directory and run the following command to associate and disassociate an EIP.

```
python  natgw_asso  ciate_eip . py
```

The system displays the following output:

```
--------------------------- create_vpc ---------------------------
{
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RouteTable  Id ": " vtb - uf6agemvkc  md8xxxxxx  x ",
  " VRouterId ": " vrt - uf6r7lqtsv  65dxxxxxx  x ",
  " VpcId ": " vpc - uf6mqfqx8v  jmoxxxxxx  x ",
  " RequestId ": " ADF806C6 - FCD6 - 4E46 - B8E3 - 72C2BE8953  44 "
}

--------------------------- create_vsw  itch
 ---------------------------
{
  " VSwitchId ": " vsw - uf6rm6add6  w89xxxxxx  x ",
  " RequestId ": " 897FFCC1 - E6BA - 484E - A245 - C5DAEBBA26  9C "
}

--------------------------- create_nat  _gateway
 ---------------------------
{
  " NatGateway  Id ": " ngw - uf681h38pb  vlyxxxxxx  x ",
  " BandwidthP  ackageIds ": {
    " BandwidthP  ackageId ": []
  },
  " ForwardTab  leIds ": {
    " ForwardTab  leId ": [
      " ftb - uf6jd0vbya  o2dxxxxxx  x "
    ]
  },
  " RequestId ": " 7C7CD3CB - 041A - 4B80 - 80B4 - 8BF8D5EF0D  26 ",
  " SnatTableI  ds ": {
    " SnatTableI  d ": [
      " stb - uf6uj997ht  g3uxxxxxx  x "
    ]
  }
}

--------------------------- allocate_e  ip_address
 ---------------------------
{
  " EipAddress ": " 106 . xx . xx . 129 ",
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RequestId ": " DB795B99 - 1CEA - 4FC1 - 9CE3 - 9DE2B977BF  02 ",
  " Allocation  Id ": " eip - uf62tf8y4u  yacxxxxxx  x "
}

--------------------------- associate_  eip_addres  s   eip
 ---------------------------
{
  " RequestId ": " 443D7060 - B716 - 4193 - A44D - 23FE762004  F8 "
}
```

```
--------------------------- describe_e  ip_address
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " PageSize ":  10 ,
  " EipAddress  es ": {
    " EipAddress ": [
      {
        " ISP ": " BGP ",
        " ExpiredTim  e ": "",
        " InternetCh  argeType ": " PayByBandw  idth ",
        " IpAddress ": " 106 . xx . xx . 129 ",
        " Allocation  Id ": " eip - uf62tf8y4u  yacxxxxxxx  x ",
        " PrivateIpA  ddress ": "",
        " Status ": " InUse ",
        " BandwidthP  ackageId ": "",
        " InstanceId ": " ngw - uf681h38pb  vlyxxxxxxx  x ",
        " InstanceRe  gionId ": " cn - shanghai ",
        " RegionId ": " cn - shanghai ",
        " AvailableR  egions ": {
          " AvailableR  egion ": [
            " cn - shanghai "
          ]
        },
        " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
        " HasReserva  tionData ":  false ,
        " InstanceTy  pe ": " Nat ",
        " Allocation  Time ": " 2019 - 04 - 24T10 : 03 : 08Z ",
        " Name ": "",
        " OperationL  ocks ": {
          " LockReason ": []
        },
        " Mode ": " NAT ",
        " BandwidthP  ackageType ": "",
        " BandwidthP  ackageBand  width ": "",
        " Bandwidth ": " 5 ",
        " HDMonitorS  tatus ": " OFF ",
        " ChargeType ": " PostPaid ",
        " SecondLimi  ted ":  false ,
        " Descritpio  n ": ""
      }
    ]
  },
  " RequestId ": " F0AEE605 - 14AD - 4ADD - 980C - 4B508CE7EE  4B "
}

--------------------------- create_com  mon_bandwi  dth_packag  e
 ----------------------
-----
{
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " BandwidthP  ackageId ": " cbwp - uf6dmfvq0g  zzgxxxxxxx  x ",
  " RequestId ": " D5E02777 - 2A72 - 42EC - 8308 - 5D4B6E56D9  00 "
}

--------------------------- add_common  _bandwidth  _packageIp
 ----------------------
----
{
  " RequestId ": " 3B4CD99C - 6E59 - 4DA2 - 9256 - EACF3F6994  12 "
}
```

```
--------------------------- describe_n  at_gateway
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " RequestId ": " A07498A0 - 4D60 - 4E0F - A7DE - 3A832B174D  59 ",
  " PageSize ":  10 ,
  " NatGateway  s ": {
    " NatGateway ": [
      {
        " Status ": " Available ",
        " BandwidthP  ackageIds ": {
          " BandwidthP  ackageId ": []
        },
        " VpcId ": " vpc - uf6mqfqx8v  jmoxxxxxxx  x ",
        " Descriptio  n ": "",
        " ForwardTab  leIds ": {
          " ForwardTab  leId ": [
            " ftb - uf6jd0vbya  o2dxxxxxxx  x "
          ]
        },
        " IpLists ": {
          " IpList ": [
            {
              " UsingStatu  s ": " Idle ",
              " IpAddress ": " 106 . xx . xx . 129 ",
              " Allocation  Id ": " eip - uf62tf8y4u  yacxxxxxxx  x
 "
            }
          ]
        },
        " BusinessSt  atus ": " Normal ",
        " RegionId ": " cn - shanghai ",
        " CreationTi  me ": " 2019 - 04 - 24T10 : 03 : 05Z ",
        " NatGateway  Id ": " ngw - uf681h38pb  vlyxxxxxxx  x ",
        " SnatTableI  ds ": {
          " SnatTableI  d ": [
            " stb - uf6uj997ht  g3uxxxxxxx  x "
          ]
        },
        " AutoPay ":  false ,
        " InstanceCh  argeType ": " PostPaid ",
        " ExpiredTim  e ": "",
        " Spec ": " Small ",
        " Name ": ""
      }
    ]
  }
}

--------------------------- unassociat  e_eip_addr  ess   nat
 ---------------------------
-
{
  " RequestId ": " 92CB670E - 239D - 4659 - B91F - E0565D5C0F  2D "
}

--------------------------- remove_com  mon_bandwi  dth_packag  eIp
 --------------------
-------
{
  " RequestId ": " A58E9647 - 6761 - 4CA3 - 8786 - 3CD4E7D2A7  AB "
}
```

```
--------------------------- delete_com  mon_bandwi  dth_packag  e
 ----------------------
-----
{
  " RequestId ": " 4AA428BC - B72F - 4567 - 94B8 - AC2398EF75  29 "
}

--------------------------- delete_nat  _gateway
 ---------------------------
{
  " RequestId ": " EC6C5D04 - AF7D - 4560 - A30E - 80EC141D17  4D "
}

--------------------------- release_ei  p_address
 ---------------------------
{
  " RequestId ": " 9B1380B3 - EE97 - 49BD - 88FE - DBF3563042  08 "
}

--------------------------- delete_vsw  itch
 ---------------------------
{
  " RequestId ": " A9A1D63E - 5709 - 4B98 - 90BF - 9069AA2642  30 "
}

--------------------------- delete_vpc --------------------------
{
  " RequestId ": " 3B687C37 - 5315 - 4E0B - BE13 - 103BB287A8  0D "
}
```

2.5.3 Create a DNAT entry

This topic describes how to use the Alibaba Cloud Python SDK to create a DNAT entry.

Context

The detailed steps used in this tutorial to create a DNAT entry are as follows:

1. Create a VPC in the China (Shanghai) region.
2. Create a VSwitch under the VPC.
3. Create a NAT Gateway under the VPC.
4. Create an EIP in the China (Shanghai) region.
5. Attach the EIP to the NAT Gateway.
6. Create a DNAT entry.
7. Query the EIP attached to the NAT Gateway.
8. Query the NAT Gateway.
9. Delete the DNAT entry.
10. Detach the EIP from the NAT Gateway.
11. Delete the NAT Gateway.
12. Release the EIP.
13. Delete the VSwitch.

14.Delete the VPC.

Procedure

1. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* folder.
2. Open the *consts . py* file in your text editor and configure the ACS_CLIENT parameter used for user authentication.
3. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ natgw* folder.
4. Open the *natgw_dnat . py* file in your text editor and configure your required parameters. Then, save the configurations and exit the text editor.

```
# encoding = utf - 8
 import   sys
 import   json
 import   time

 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   CreateNatG
 atewayRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DeleteNatG
 atewayRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DescribeNa
 tGatewaysR  equest
 from   aliyunsdkv  pc . request . v20160428   import   CreateForw
 ardEntryRe  quest
 from   aliyunsdkv  pc . request . v20160428   import   DescribeFo
 rwardTable  EntriesReq  uest
 from   aliyunsdkv  pc . request . v20160428   import   DeleteForw
 ardEntryRe  quest
 from   sdk_lib . sdk_vpc   import   Vpc
 from   sdk_lib . sdk_vswitc  h   import   VSwitch
 from   sdk_lib . sdk_eip   import   Eip
 from   sdk_lib . sdk_cbwp   import   CommonBand  widthPacka  ge
 from   sdk_lib . common_uti  l   import   CommonUtil
 from   sdk_lib . check_stat  us   import   CheckStatu  s
 from   sdk_lib . exception   import   ExceptionH  andler
 from   sdk_lib . consts   import  *

 client  =  ACS_CLIENT


 class   NatGateway ( object ):
     def   __init__ ( self ,  client ):
         self . client  =  client

     def   create_nat  _gateway ( self ,  params ):
        """
         create_nat  _gateway :  Create   a   NAT   Gateway
         API   reference   link :  https :// www . alibabaclo  ud .
 com / help / doc - detail / 36048 . htm
        """
         try :
```

```
            request = CreateNatG atewayRequ est . CreateNatG
atewayRequ est ()
            request . set_VpcId ( params [' vpc_id '])
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
          # Check if the NAT Gateway is in the
available state
            if CheckStatu s . check_stat us ( TIME_DEFAU
LT_OUT , DEFAULT_TI ME ,
                                        self . describe_n
at_gateway _status ,
                                        AVAILABLE , response_j
son [' NatGateway Id ']):
                return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_n at_gateway ( self , nat_gatewa y_id ):
        """
        describe_n at_gateway : Query NAT Gateways in a
region
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36054 . htm
        """
        try :
            request = DescribeNa tGatewaysR equest .
DescribeNa tGatewaysR equest ()
            request . set_NatGat ewayId ( nat_gatewa y_id )
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def delete_nat _gateway ( self , params ):
        """
        delete_nat _gateway : Delete a NAT Gateway
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36051 . htm
        """
        try :
            request = DeleteNatG atewayRequ est . DeleteNatG
atewayRequ est ()
            request . set_NatGat ewayId ( params [' nat_gatewa
y_id '])
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
          # Check if the NAT Gateway is in the
available state
            if CheckStatu s . check_stat us ( TIME_DEFAU
LT_OUT , DEFAULT_TI ME * 5 ,
                                        self . describe_n
at_gateway _status ,
                                       '', params [' nat_gatewa
y_id ']):
                return response_j son
        except ServerExce ption as e :
```

```
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_n  at_gateway  _status ( self ,  nat_gatewa
y_id ):
        """
        describe_n  at_gateway  _status :  Query   the   status
of   NAT   Gateways   in   a   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36054 . htm
        """
        response  =  self . describe_n  at_gateway ( nat_gatewa
y_id )
        if   len ( response [" NatGateway  s "][" NatGateway "])
==  0 :
            return  ''
        return   response [" NatGateway  s "][" NatGateway "][ 0 ]
[' Status ']

    def   create_for  ward_entry ( self ,  params ):
        """
        create_for  ward_entry :  Create   a   forward   entry
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36058 . htm
        """
        try :
            request  =  CreateForw  ardEntryRe  quest . CreateForw
ardEntryRe  quest ()
            request . set_Forwar  dTableId ( params [' forward_ta
ble_id '])
            request . set_Extern  alIp ( params [' external_i  p
'])
            request . set_IpProt  ocol ( params [' ip_protoco  l
'])
            request . set_Extern  alPort ( params [' external_p
ort '])
            request . set_Intern  alIp ( params [' internal_i  p
'])
            request . set_Intern  alPort ( params [' internal_p
ort '])
            response  =  client . do_action_  with_excep  tion (
request )
            response_j  son  =  json . loads ( response )
          #  Check   if   the   forward   entry   is   in   the
available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                        self . describe_f
orward_sta  tus ,
                                        AVAILABLE ,  params ['
forward_ta  ble_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_f  orward ( self ,  forward_ta  ble_id ):
        """
        describe_f  orward :  Query   DNAT   entries   in   a
region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36053 . htm
```

```
        """
        try :
            request = DescribeFo rwardTable EntriesReq uest .
DescribeFo rwardTable EntriesReq uest ()
            request . set_Forwar dTableId ( forward_ta ble_id )
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_f orward_sta tus ( self , forward_ta ble_id
):
        """
        describe_f orward_sta tus : Query the status of
DNAT entries in a region
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36053 . htm
        """
        response = self . describe_f orward ( forward_ta ble_id
)
        if len ( response [" ForwardTab leEntries "]["
ForwardTab leEntry "]) == 0 :
            return ''
        return response [" ForwardTab leEntries "][" ForwardTab
leEntry "][ 0 ][' Status ']

    def delete_for ward_entry ( self , params ):
        """
        delete_for ward_entry : Delete a forward entry
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36050 . htm
        """
        try :
            request = DeleteForw ardEntryRe quest . DeleteForw
ardEntryRe quest ()
            request . set_Forwar dTableId ( params [' forward_ta
ble_id '])
            request . set_Forwar dEntryId ( params [' forward_en
try_id '])
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
          # Check if the forward entry is in the
available state
            if CheckStatu s . check_stat us ( TIME_DEFAU
LT_OUT , DEFAULT_TI ME * 5 ,
                                        self . describe_f
orward_sta tus ,
                                       '', params [' forward_ta
ble_id ']):
                return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )


def main ():
    vpc = Vpc ( client )
    vswitch = VSwitch ( client )
```

```
    eip  =  Eip ( client )
    cbwp  =  CommonBand  widthPacka  ge ( client )
    nat_gatewa  y  =  NatGateway ( client )

    params  = {}

   #  Create   a   VPC
    vpc_json  =  vpc . create_vpc ()
    CommonUtil . log (" create_vpc ",  vpc_json )

   #  Create   a   VSwitch
    params [' vpc_id '] =  vpc_json [' VpcId ']
    params [' zone_id '] = " cn - shanghai - b "
    params [' cidr_block '] = " 172 . 16 . 1 . 0 / 24 "
    vswitch_js  on  =  vswitch . create_vsw  itch ( params )
    CommonUtil . log (" create_vsw  itch ",  vswitch_js  on )
    params [' vswitch_id '] =  vswitch_js  on [' VSwitchId ']

   #  Create   a   NAT   Gateway
    nat_gatewa  y_json  =  nat_gatewa  y . create_nat  _gateway (
params )
    CommonUtil . log (" create_nat  _gateway ",  nat_gatewa  y_json
)

   #  Create   an   EIP
    eip_respon  se_json  =  eip . allocate_e  ip_address ( params )
    CommonUtil . log (" allocate_e  ip_address ",  eip_respon
se_json )
    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]
    params [' external_i  p '] =  eip_respon  se_json [' EipAddress
']

   #  Attach   the   EIP   to   the   NAT   Gateway
    params [' instance_i  d '] =  nat_gatewa  y_json [' NatGateway
Id ']
    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]
    params [' instance_t  ype '] = ' Nat '
    eip_respon  se_json  =  eip . associate_  eip_addres  s (
params )
    CommonUtil . log (" associate_  eip_addres  s   eip ",
eip_respon  se_json )

   #  Create   a   DNAT   entry
    params [' forward_ta  ble_id '] =  nat_gatewa  y_json ['
ForwardTab  leIds '][' ForwardTab  leId '][ 0 ]
    params [' ip_protoco  l '] = ' tcp '
    params [' external_p  ort '] = ' 8080 '
    params [' internal_p  ort '] = ' 80 '
    params [' internal_i  p '] = ' 172 . 16 . 1 . 0 '
    forward_en  try_json  =  nat_gatewa  y . create_for  ward_entry
( params )
    CommonUtil . log (" create_for  ward_entry ",  forward_en
try_json )

   #  Query   the   EIP
    eip_respon  se_json  =  eip . describe_e  ip_address ( params
[' allocation  _id '])
    CommonUtil . log (" describe_e  ip_address ",  eip_respon
se_json )

   #  Query   the   NAT   Gateway
```

```
    params [' nat_gatewa  y_id '] =  nat_gatewa  y_json ['
NatGateway  Id ']
    nat_gatewa  y_json  =  nat_gatewa  y . describe_n  at_gateway (
params [' nat_gatewa  y_id '])
    CommonUtil . log (" describe_n  at_gateway ",  nat_gatewa
y_json )

   #  Delete   the   DNAT   entry
    params [' forward_en  try_id '] =  forward_en  try_json ['
ForwardEnt  ryId ']
    forward_en  try_json  =  nat_gatewa  y . delete_for  ward_entry
( params )
    CommonUtil . log (" delete_for  ward_entry ",  forward_en
try_json )

   #  Detach   the   EIP
    eip_respon  se_json  =  eip . unassociat  e_eip_addr  ess (
params )
    CommonUtil . log (" unassociat  e_eip_addr  ess   nat ",
eip_respon  se_json )

   #  Delete   the   NAT   Gateway
    nat_gatewa  y_json  =  nat_gatewa  y . delete_nat  _gateway (
params )
    CommonUtil . log (" delete_nat  _gateway ",  nat_gatewa  y_json
)

   #  Release   the   EIP
    eip_respon  se_json  =  eip . release_ei  p_address ( params )
    CommonUtil . log (" release_ei  p_address ",  eip_respon
se_json )

   #  Delete   the   VSwitch
    params [' vswitch_id '] =  vswitch_js  on [' VSwitchId ']
    vswitch_js  on  =  vswitch . delete_vsw  itch ( params )
    CommonUtil . log (" delete_vsw  itch ",  vswitch_js  on )

   #  Delete   the   VPC
    vpc_json  =  vpc . delete_vpc ( params )
    CommonUtil . log (" delete_vpc ",  vpc_json )


if   __name__  == " __main__ ":
    sys . exit ( main ())
```

5. **Access the** *natgw_dnat . py* **directory and run the following command to create a DNAT entry.**

```
python   natgw_dnat . py
```

The system displays the following output:

```
--------------------------- create_vpc ---------------------------
{
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RouteTable  Id ": " vtb - uf63rln6gb  b50xxxxxxx  x ",
  " VRouterId ": " vrt - uf6p1hfo0h  o8gxxxxxxx  x ",
  " VpcId ": " vpc - uf6c3r8yca  7dhxxxxxxx  x ",
  " RequestId ": " 1F97FC59 - 77DF - 4D76 - BE62 - 0A13EB4E61  4C "
}
```

```
--------------------------- create_vsw  itch
 ---------------------------
{
  " VSwitchId ": " vsw - uf6liy66d9  ssuxxxxxxx  x ",
  " RequestId ": " 88CCCFED - 1448 - 49D2 - 8550 - 71952981A4  7A "
}

--------------------------- create_nat  _gateway
 ---------------------------
{
  " NatGateway  Id ": " ngw - uf6aolgwhs  svsxxxxxxx  x ",
  " BandwidthP  ackageIds ": {
    " BandwidthP  ackageId ": []
  },
  " ForwardTab  leIds ": {
    " ForwardTab  leId ": [
      " ftb - uf6unjiun4  i12xxxxxxx  x "
    ]
  },
  " RequestId ": " 62A58351 - D608 - 43A4 - 849E - 1E177E917B  EA ",
  " SnatTableI  ds ": {
    " SnatTableI  d ": [
      " stb - uf65utljwc  dkpxxxxxxx  x "
    ]
  }
}

--------------------------- allocate_e  ip_address
 ---------------------------
{
  " EipAddress ": " 101 . xx . xx . 110 ",
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RequestId ": " 0565295E - 2F49 - 4511 - 93BC - 747A2D19A6  BD ",
  " Allocation  Id ": " eip - uf683xrl32  ge8xxxxxxx  x "
}

--------------------------- associate_  eip_addres  s   eip
 ---------------------------
{
  " RequestId ": " 8759FCE8 - F8C2 - 4372 - 91D5 - 7A25D43FD7  8C "
}

--------------------------- create_for  ward_entry
 ---------------------------
{
  " ForwardEnt  ryId ": " fwd - uf6ng3wt8s  fwmxxxxxxx  x ",
  " RequestId ": " CC81BCF6 - 2F64 - 40CF - 85B0 - 676A83AC39  02 "
}

--------------------------- describe_e  ip_address
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " PageSize ":  10 ,
  " EipAddress  es ": {
    " EipAddress ": [
      {
        " ISP ": " BGP ",
        " ExpiredTim  e ": "",
        " InternetCh  argeType ": " PayByBandw  idth ",
        " IpAddress ": " 101 . xx . xx . 110 ",
        " Allocation  Id ": " eip - uf683xrl32  ge8xxxxxxx  x ",
        " PrivateIpA  ddress ": "",
```

```
        " Status ": " InUse ",
        " BandwidthP  ackageId ": "",
        " InstanceId ": " ngw - uf6aolgwhs  svsxxxxxxx  x ",
        " InstanceRe  gionId ": " cn - shanghai ",
        " RegionId ": " cn - shanghai ",
        " AvailableR  egions ": {
          " AvailableR  egion ": [
            " cn - shanghai "
          ]
        },
        " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
        " HasReserva  tionData ":  false ,
        " InstanceTy  pe ": " Nat ",
        " Allocation  Time ": " 2019 - 04 - 24T10 : 56 : 53Z ",
        " Name ": "",
        " OperationL  ocks ": {
          " LockReason ": []
        },
        " Mode ": " NAT ",
        " BandwidthP  ackageType ": "",
        " BandwidthP  ackageBand  width ": "",
        " Bandwidth ": " 5 ",
        " HDMonitorS  tatus ": " OFF ",
        " ChargeType ": " PostPaid ",
        " SecondLimi  ted ":  false ,
        " Descritpio  n ": ""
      }
    ]
  },
  " RequestId ": " CD2B3613 - 2A99 - 4687 - 9C23 - A8E9F1F030  48 "
}

--------------------------- describe_n  at_gateway
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " RequestId ": " D7519663 - 8D3B - 4CC5 - 894F - A6798C8968  8D ",
  " PageSize ":  10 ,
  " NatGateway  s ": {
    " NatGateway ": [
      {
        " Status ": " Available ",
        " BandwidthP  ackageIds ": {
          " BandwidthP  ackageId ": []
        },
        " VpcId ": " vpc - uf6c3r8yca  7dhxxxxxxx  x ",
        " Descriptio  n ": "",
        " ForwardTab  leIds ": {
          " ForwardTab  leId ": [
            " ftb - uf6unjiun4  i12xxxxxxx  x "
          ]
        },
        " IpLists ": {
          " IpList ": [
            {
              " UsingStatu  s ": " UsedByForw  ardTable ",
              " IpAddress ": " 101 . xx . xx . 110 ",
              " Allocation  Id ": " eip - uf683xrl32  ge8xxxxxxx  x
 "
            }
          ]
        },
        " BusinessSt  atus ": " Normal ",
```

```
        " RegionId ": " cn - shanghai ",
        " CreationTi  me ": " 2019 - 04 - 24T10 : 56 : 50Z ",
        " NatGateway  Id ": " ngw - uf6aolgwhs  svsxxxxxxx  x ",
        " SnatTableI  ds ": {
          " SnatTableI  d ": [
            " stb - uf65utljwc  dkpxxxxxxx  x "
          ]
        },
        " AutoPay ":  false ,
        " InstanceCh  argeType ": " PostPaid ",
        " ExpiredTim  e ": "",
        " Spec ": " Small ",
        " Name ": ""
      }
    ]
  }
}

--------------------------- delete_for  ward_entry
 ---------------------------
{
  " RequestId ": " 32C76D08 - 5738 - 4B07 - A638 - ACE5F5F522  0E "
}

--------------------------- unassociat  e_eip_addr  ess   nat
 -------------------------
_
{
  " RequestId ": " AE686920 - 2CD1 - 4850 - AADC - C249484D4B  1A "
}

--------------------------- delete_nat  _gateway
 ---------------------------
{
  " RequestId ": " FEBB1E7A - BA5B - 4445 - B2AB - 5B828C17BB  E6 "
}

--------------------------- release_ei  p_address
 ---------------------------
{
  " RequestId ": " 812D5E78 - 5113 - 4B92 - 892D - 0B293BAD66  F6 "
}

--------------------------- delete_vsw  itch
 ---------------------------
{
  " RequestId ": " 8E13EEE4 - 21B5 - 4280 - B46B - 5C168736DC  3A "
}

--------------------------- delete_vpc ---------------------------
{
  " RequestId ": " DCBA91E7 - F355 - 4EB6 - 83E3 - 27F2E68A84  35 "
}
```

2.5.4 Create an SNAT entry

This topic describes how to use the Alibaba Cloud Python SDK to create an SNAT entry.

Context

The detailed steps used in this tutorial to create an SNAT entry are as follows:

1. Create a VPC in the China (Shanghai) region.
2. Create a VSwitch under the VPC.
3. Create a NAT Gateway under the VPC.
4. Create an EIP in the China (Shanghai) region.
5. Attach the EIP to the NAT Gateway.
6. Create an SNAT entry.
7. Query the EIP attached to the NAT Gateway.
8. Query the NAT Gateway.
9. Delete the SNAT entry.
10. Detach the EIP from the NAT Gateway.
11. Delete the NAT Gateway.
12. Release the EIP.
13. Delete the VSwitch.
14. Delete the VPC.

Procedure

1. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* folder.
2. Open the *consts . py* file in your text editor and configure the ACS_CLIENT parameter used for user authentication.
3. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ natgw* folder.
4. Open the *natgw_snat . py* file in your text editor and configure your required parameters. Then, save the configurations and exit the text editor.

```
# encoding = utf - 8
 import   sys
 import   json
 import   time

 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   CreateNatG
 atewayRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DeleteNatG
 atewayRequ  est
 from   aliyunsdkv  pc . request . v20160428   import   DescribeNa
 tGatewaysR  equest
 from   aliyunsdkv  pc . request . v20160428   import   CreateSnat
 EntryReque  st
 from   aliyunsdkv  pc . request . v20160428   import   DescribeSn
 atTableEnt  riesReques  t
```

```
from aliyunsdkv pc . request . v20160428 import DeleteSnat
EntryReque st
from sdk_lib . sdk_vpc import Vpc
from sdk_lib . sdk_vswitc h import VSwitch
from sdk_lib . sdk_eip import Eip
from sdk_lib . sdk_cbwp import CommonBand widthPacka ge
from sdk_lib . common_uti l import CommonUtil
from sdk_lib . check_stat us import CheckStatu s
from sdk_lib . exception import ExceptionH andler
from sdk_lib . consts import *

client = ACS_CLIENT


class NatGateway ( object ):
    def __init__ ( self , client ):
        self . client = client

    def create_nat _gateway ( self , params ):
       """
        create_nat _gateway : Create a NAT Gateway
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36048 . htm
       """
        try :
            request = CreateNatG atewayRequ est . CreateNatG
atewayRequ est ()
            request . set_VpcId ( params [' vpc_id '])
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
           # Check if the NAT Gateway is in the
available state
            if CheckStatu s . check_stat us ( TIME_DEFAU
LT_OUT , DEFAULT_TI ME ,
                                        self . describe_n
at_gateway _status ,
                                        AVAILABLE , response_j
son [' NatGateway Id ']):
                return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_n at_gateway ( self , nat_gatewa y_id ):
       """
        describe_n at_gateway : Query NAT Gateways in a
region
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36054 . htm
       """
        try :
            request = DescribeNa tGatewaysR equest .
DescribeNa tGatewaysR equest ()
            request . set_NatGat ewayId ( nat_gatewa y_id )
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
            return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )
```

```
    def delete_nat _gateway ( self , params ):
        """
        delete_nat _gateway : Delete the NAT Gateway
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36051 . htm
        """
        try :
            request = DeleteNatG atewayRequ est . DeleteNatG
atewayRequ est ()
            request . set_NatGat ewayId ( params [' nat_gatewa
y_id '])
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
           # Check if the NAT Gateway is in the
available state
            if CheckStatu s . check_stat us ( TIME_DEFAU
LT_OUT , DEFAULT_TI ME * 5 ,
                                        self . describe_n
at_gateway _status ,
                                       '', params [' nat_gatewa
y_id ']):
                return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def describe_n at_gateway _status ( self , nat_gatewa
y_id ):
        """
        describe_n at_gateway _status : Query the status
of NAT Gateways in a region
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36054 . htm
        """
        response = self . describe_n at_gateway ( nat_gatewa
y_id )
        if len ( response [" NatGateway s "][" NatGateway "])
== 0 :
            return ''
        return response [" NatGateway s "][" NatGateway "][ 0 ]
[' Status ']

    def create_sna t_entry ( self , params ):
        """
        describe_s nat : Create an SNAT entry
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 42672 . htm
        """
        try :
            request = CreateSnat EntryReque st . CreateSnat
EntryReque st ()
            request . set_SnatTa bleId ( params [' snat_table _id
'])
            request . set_Source VSwitchId ( params [' vswitch_id
'])
            request . set_SnatIp ( params [' snat_ip '])
            response = client . do_action_ with_excep tion (
request )
            response_j son = json . loads ( response )
           # Check if the SNAT entry is in the
available state
```

```
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                        self . describe_s
nat_status ,
                                        AVAILABLE ,  params ['
snat_table  _id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_s  nat ( self ,  snat_table  _id ):
        """
        describe_s  nat :  Query   SNAT   entries   in   a
region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 42677 . htm
        """
        try :
            request  =  DescribeSn  atTableEnt  riesReques  t .
DescribeSn  atTableEnt  riesReques  t ()
            request . set_SnatTa  bleId ( snat_table  _id )
            response  =  client . do_action_  with_excep  tion (
request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_s  nat_status ( self ,  snat_table  _id ):
        """
        describe_s  nat_status :  Query   the   status   of   SNAT
  entries   in   a   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 42677 . htm
        """
        response  =  self . describe_s  nat ( snat_table  _id )
        if   len ( response [" SnatTableE  ntries "][" SnatTableE
ntry "]) ==  0 :
            return  ''
        return   response [" SnatTableE  ntries "][" SnatTableE
ntry "][ 0 ][' Status ']

    def   delete_sna  t_entry ( self ,  params ):
        """
        delete_sna  t_entry :  Delete   an   SNAT   entry
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 42678 . htm
        """
        try :
            request  =  DeleteSnat  EntryReque  st . DeleteSnat
EntryReque  st ()
            request . set_SnatTa  bleId ( params [' snat_table  _id
'])
            request . set_SnatEn  tryId ( params [' snat_entry  _id
'])
            response  =  client . do_action_  with_excep  tion (
request )
            response_j  son  =  json . loads ( response )
           #  Check   if   the   SNAT   entry   is   in   the
available   state
```

```
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME  *  5 ,
                                        self . describe_s
nat_status ,
                                       '',  params [' snat_table
_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )


def   main ():
    vpc  =  Vpc ( client )
    vswitch  =  VSwitch ( client )
    eip  =  Eip ( client )
    cbwp  =  CommonBand  widthPacka  ge ( client )
    nat_gatewa  y  =  NatGateway ( client )

    params  = {}

   #  Create   a   VPC
    vpc_json  =  vpc . create_vpc ()
    CommonUtil . log (" create_vpc ",  vpc_json )

   #  Create   a   VSwitch
    params [' vpc_id '] =  vpc_json [' VpcId ']
    params [' zone_id '] = " cn - shanghai - b "
    params [' cidr_block '] = " 172 . 16 . 1 . 0 / 24 "
    vswitch_js  on  =  vswitch . create_vsw  itch ( params )
    CommonUtil . log (" create_vsw  itch ",  vswitch_js  on )
    params [' vswitch_id '] =  vswitch_js  on [' VSwitchId ']

   #  Create   a   NAT   Gateway
    nat_gatewa  y_json  =  nat_gatewa  y . create_nat  _gateway (
params )
    CommonUtil . log (" create_nat  _gateway ",  nat_gatewa  y_json
)

   #  Create   an   EIP
    eip_respon  se_json  =  eip . allocate_e  ip_address ( params )
    CommonUtil . log (" allocate_e  ip_address ",  eip_respon
se_json )
    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]
    params [' snat_ip '] =  eip_respon  se_json [' EipAddress ']

   #  Attach   the   EIP   to   the   NAT   Gateway
    params [' instance_i  d '] =  nat_gatewa  y_json [' NatGateway
Id ']
    params [' allocation  _id '] =  eip_respon  se_json ["
Allocation  Id "]
    params [' instance_t  ype '] = ' Nat '
    eip_respon  se_json  =  eip . associate_  eip_addres  s (
params )
    CommonUtil . log (" associate_  eip_addres  s   eip ",
eip_respon  se_json )

   #  Create   an   SNAT   entry
    params [' snat_table  _id '] =  nat_gatewa  y_json ['
SnatTableI  ds '][' SnatTableI  d '][ 0 ]
    snat_entry  _json  =  nat_gatewa  y . create_sna  t_entry (
params )
```

```
    CommonUtil . log (" create_sna  t_entry ",  snat_entry  _json )

   #  Query   the   EIP
    eip_respon  se_json  =  eip . describe_e  ip_address ( params
[' allocation  _id '])
    CommonUtil . log (" describe_e  ip_address ",  eip_respon
se_json )

   #  Query   the   NAT   Gateway
    params [' nat_gatewa  y_id '] =  nat_gatewa  y_json ['
NatGateway  Id ']
    nat_gatewa  y_json  =  nat_gatewa  y . describe_n  at_gateway (
params [' nat_gatewa  y_id '])
    CommonUtil . log (" describe_n  at_gateway ",  nat_gatewa
y_json )

   #  Delete   the   SNAT   entry
    params [' snat_entry  _id '] =  snat_entry  _json ['
SnatEntryI  d ']
    snat_entry  _json  =  nat_gatewa  y . delete_sna  t_entry (
params )
    CommonUtil . log (" delete_sna  t_entry ",  snat_entry  _json )

   #  Detach   the   EIP
    eip_respon  se_json  =  eip . unassociat  e_eip_addr  ess (
params )
    CommonUtil . log (" unassociat  e_eip_addr  ess   nat ",
eip_respon  se_json )

   #  Delete   the   NAT   Gateway
    nat_gatewa  y_json  =  nat_gatewa  y . delete_nat  _gateway (
params )
    CommonUtil . log (" delete_nat  _gateway ",  nat_gatewa  y_json
)

   #  Release   the   EIP
    eip_respon  se_json  =  eip . release_ei  p_address ( params )
    CommonUtil . log (" release_ei  p_address ",  eip_respon
se_json )

   #  Delete   the   VSwitch
    params [' vswitch_id '] =  vswitch_js  on [' VSwitchId ']
    vswitch_js  on  =  vswitch . delete_vsw  itch ( params )
    CommonUtil . log (" delete_vsw  itch ",  vswitch_js  on )

   #  Delete   the   VPC
    vpc_json  =  vpc . delete_vpc ( params )
    CommonUtil . log (" delete_vpc ",  vpc_json )


if   __name__  == " __main__ ":
    sys . exit ( main ())
```

5. **Access the** *natgw_snat . py* **directory and run the following command to create an SNAT entry.**

```
python   natgw_snat . py
```

The system displays the following output:

```
-------------------------- create_vpc --------------------------
```

```
{
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RouteTable  Id ": " vtb - uf6a8ccj9n  e58xxxxxxx  x ",
  " VRouterId ": " vrt - uf6qqqaf1o  1ptxxxxxxx  x ",
  " VpcId ": " vpc - uf6hxer3h0  7wgxxxxxxx  x ",
  " RequestId ": " 8F483A7B - 8A38 - 47ED - 85BD - 1E83C075AE  A4 "
}

--------------------------- create_vsw  itch
 ---------------------------
{
  " VSwitchId ": " vsw - uf6lbov9ty  etqxxxxxxx  x ",
  " RequestId ": " 2EE2E11B - EF60 - 4C88 - BE2A - F45517290B  31 "
}

--------------------------- create_nat  _gateway
 ---------------------------
{
  " NatGateway  Id ": " ngw - uf6l3c3rsw  ubuxxxxxxx  x ",
  " BandwidthP  ackageIds ": {
    " BandwidthP  ackageId ": []
  },
  " ForwardTab  leIds ": {
    " ForwardTab  leId ": [
      " ftb - uf6086r1hy  ecbxxxxxxx  x "
    ]
  },
  " RequestId ": " 9037D769 - 24C8 - 46AD - 83F3 - 4C0538FA59  70 ",
  " SnatTableI  ds ": {
    " SnatTableI  d ": [
      " stb - uf6ppo11rs  ecmxxxxxxx  x "
    ]
  }
}

--------------------------- allocate_e  ip_address
 ---------------------------
{
  " EipAddress ": " 101 . xx . xx . 110 ",
  " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
  " RequestId ": " 0DE621B4 - 6BDE - 4E17 - A294 - 8F71FBB9F7  10 ",
  " Allocation  Id ": " eip - uf6d311cpm  r0nxxxxxxx  x "
}

--------------------------- associate_  eip_addres  s   eip
 ---------------------------
{
  " RequestId ": " C95B2EDC - F081 - 4784 - B60B - 2600F60E68  4D "
}

--------------------------- create_sna  t_entry
 ---------------------------
{
  " SnatEntryI  d ": " snat - uf6ppbwshd  u40xxxxxxx  x ",
  " RequestId ": " BB9F8FD2 - 3CB5 - 4F84 - 8006 - FE64BF3BEA  06 "
}

--------------------------- describe_e  ip_address
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " PageSize ":  10 ,
  " EipAddress  es ": {
```

```
    " EipAddress ": [
      {
        " ISP ": " BGP ",
        " ExpiredTim  e ": "",
        " InternetCh  argeType ": " PayByBandw  idth ",
        " IpAddress ": " 101 . xx . xx . 110 ",
        " Allocation  Id ": " eip - uf6d311cpm  r0nxxxxxxx  x ",
        " PrivateIpA  ddress ": "",
        " Status ": " InUse ",
        " BandwidthP  ackageId ": "",
        " InstanceId ": " ngw - uf6l3c3rsw  ubuxxxxxxx  x ",
        " InstanceRe  gionId ": " cn - shanghai ",
        " RegionId ": " cn - shanghai ",
        " AvailableR  egions ": {
          " AvailableR  egion ": [
            " cn - shanghai "
          ]
        },
        " ResourceGr  oupId ": " rg - acfmxazxxx  xxxxx ",
        " HasReserva  tionData ":  false ,
        " InstanceTy  pe ": " Nat ",
        " Allocation  Time ": " 2019 - 04 - 24T11 : 20 : 09Z ",
        " Name ": "",
        " OperationL  ocks ": {
          " LockReason ": []
        },
        " Mode ": " NAT ",
        " BandwidthP  ackageType ": "",
        " BandwidthP  ackageBand  width ": "",
        " Bandwidth ": " 5 ",
        " HDMonitorS  tatus ": " OFF ",
        " ChargeType ": " PostPaid ",
        " SecondLimi  ted ":  false ,
        " Descritpio  n ": ""
      }
    ]
  },
  " RequestId ": " 19052237 - 6E84 - 4258 - 89B9 - 05772C33C0  DC "
}

--------------------------- describe_n  at_gateway
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " RequestId ": " 26CAA3FE - B400 - 4522 - 9582 - 2DAAF69129  AE ",
  " PageSize ":  10 ,
  " NatGateway  s ": {
    " NatGateway ": [
      {
        " Status ": " Available ",
        " BandwidthP  ackageIds ": {
          " BandwidthP  ackageId ": []
        },
        " VpcId ": " vpc - uf6hxer3h0  7wgxxxxxx  x ",
        " Descriptio  n ": "",
        " ForwardTab  leIds ": {
          " ForwardTab  leId ": [
            " ftb - uf6086r1hy  ecbxxxxxx  x "
          ]
        },
        " IpLists ": {
          " IpList ": [
            {
```

```
              " UsingStatu  s ": " UsedBySnat  Table ",
              " IpAddress ": " 101 . xx . xx . 110 ",
              " Allocation  Id ": " eip - uf6d311cpm  r0nxxxxxxx  x
 "
            }
          ]
        },
        " BusinessSt  atus ": " Normal ",
        " RegionId ": " cn - shanghai ",
        " CreationTi  me ": " 2019 - 04 - 24T11 : 20 : 06Z ",
        " NatGateway  Id ": " ngw - uf6l3c3rsw  ubuxxxxxxx  x ",
        " SnatTableI  ds ": {
          " SnatTableI  d ": [
            " stb - uf6ppo11rs  ecmxxxxxxx  x "
          ]
        },
        " AutoPay ":  false ,
        " InstanceCh  argeType ": " PostPaid ",
        " ExpiredTim  e ": "",
        " Spec ": " Small ",
        " Name ": ""
      }
    ]
  }
}

--------------------------- delete_sna  t_entry
 ---------------------------
{
  " RequestId ": " CDA82881 - ACF0 - 4DD1 - 887B - A764F56F18  0D "
}

--------------------------- unassociat  e_eip_addr  ess   nat
 -------------------------
_
{
  " RequestId ": " 140C46FF - 0DB0 - 47F9 - B0F4 - 3459DF117E  AF "
}

--------------------------- delete_nat  _gateway
 ---------------------------
{
  " RequestId ": " 8709747C - 6786 - 443C - 8AEA - 51647AA497  69 "
}

--------------------------- release_ei  p_address
 ---------------------------
{
  " RequestId ": " 0BC21C23 - 0FB3 - 4594 - 8B14 - 6552DF788C  93 "
}

--------------------------- delete_vsw  itch
 ---------------------------
{
  " RequestId ": " 16E840EC - E058 - 40C1 - A5BF - 8CDF672EA1  39 "
}

--------------------------- delete_vpc ---------------------------
{
  " RequestId ": " B5F18126 - FF2A - 4005 - 9973 - 3984034DF0  F4 "
```

```
}
```

2.6 Router interface

2.6.1 Interconnect two VPCs located in the same region and under the same account

This topic describes how to use the Alibaba Cloud Python SDK to interconnect two VPCs that are located in the same region and are under the same account.

Context

Note the following when you create router interfaces:

- Only one pair of interconnected router interfaces can be created between two VRouters.
- Up to five router interfaces can be created on a VRouter.
- If a router interface with overdue payment is under your account, you cannot create router interfaces. To resolve this issue, add funds to your account.
- Route entries in the same route table cannot have the same destination CIDR blocks.

The detailed steps used in this tutorial to interconnect two VPCs located in the same region and under the same account are as follows:

1. Create the initiator router interface.
2. Create the acceptor router interface.
3. Modify the initiator router interface.
4. Modify the acceptor router interface.
5. Query the initiator router interface.
6. Query the acceptor router interface.
7. Connect the initiator router interface to the acceptor router interface.
8. Create a route entry whose next hop is the initiator router interface.
9. Create a route entry whose next hop is the acceptor router interface.
10. Delete the route entry whose next hop is the acceptor router interface.
11. Delete the route entry whose next hop is the initiator router interface.
12. Deactivate the initiator router interface.
13. Deactivate the acceptor router interface.
14. Delete the initiator router interface.

15.Delete the acceptor router interface.

Procedure

1. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* folder.
2. Open the *consts . py* file in your text editor and configure the ACS_CLIENT parameter used for user authentication.
3. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ ec* folder.
4. Open the *ec_same_ac count . py* file in your text editor and configure your required parameters. Then, save the configurations and exit the text editor.

```
# encoding = utf - 8
import sys
import json
from aliyunsdkc ore . acs_except ion . exceptions import
ServerExce ption , ClientExce ption
from aliyunsdkv pc . request . v20160428 import CreateRout
erInterfac eRequest
from aliyunsdkv pc . request . v20160428 import DeleteRout
erInterfac eRequest
from aliyunsdkv pc . request . v20160428 import DescribeRo
uterInterf acesReques t
from aliyunsdkv pc . request . v20160428 import ConnectRou
terInterfa ceRequest
from aliyunsdkv pc . request . v20160428 import Deactivate
RouterInte rfaceReque st
from aliyunsdkv pc . request . v20160428 import ModifyRout
erInterfac eAttribute Request
from sdk_lib . sdk_route_ entry import RouteEntry
from sdk_lib . exception import ExceptionH andler
from sdk_lib . common_uti l import CommonUtil
from sdk_lib . check_stat us import CheckStatu s
from sdk_lib . consts import *

client = ACS_CLIENT

class RouterInte rface ( object ):
    def __init__ ( self , client ):
        self . client = client

    def create_rou ter_interf ace ( self , params ):
        """
        create_rou ter_interf ace : Create a router
interface
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36032 . htm
        """
        try :
            request = CreateRout erInterfac eRequest .
CreateRout erInterfac eRequest ()
            # The specificat ion of the router
interface
            request . set_Spec ( params [' spec '])
```

```
            #  The   role   of   the   router   interface
             request . set_Role ( params [' role '])
            #  The   ID   of   the   VRouter   associated   with
the   router   interface
             request . set_Router  Id ( params [' router_id '])
            #  The   type   of   the   VRouter   associated   with
the   router   interface
             request . set_Router  Type ( params [' router_typ  e
'])
            #  The   ID   of   the   region   to   which   the
connection   acceptor   belongs
             request . set_Opposi  teRegionId ( params ['
opposite_r  egion_id '])
            #  The   type   of   the   VRouter   associated   with
the   peer   router   interface
             request . set_Opposi  teRouterTy  pe ( params ['
opposite_r  outer_type '])

             response  =  self . client . do_action_  with_excep
tion ( request )
             response_j  son  =  json . loads ( response )
            #  Check   if   the   router   interface   is   in   the
  available   state
             if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                          self . describe_r  i_status
,
                                         ' Idle ',  response_j  son
[' RouterInte  rfaceId ']):
                 return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   connect_ro  uter_inter  face ( self ,  params ):
        """
        connect_ro  uter_inter  face :  The   initiator   router
interface   initiates   the   connection   to   the   acceptor
router   interface
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36031 . htm
        """
        try :
            request  =  ConnectRou  terInterfa  ceRequest .
ConnectRou  terInterfa  ceRequest ()
            #  The   ID   of   the   initiator   router   interface
             request . set_Router  InterfaceI  d ( params ['
router_int  erface_id '])
             response  =  self . client . do_action_  with_excep
tion ( request )
             response_j  son  =  json . loads ( response )
             if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT  *  5 ,  DEFAULT_TI  ME  *  5 ,
                                          self . describe_r  i_status
,
                                         ' Active ',  params ['
router_int  erface_id ']):
                 return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )
```

```
    def deactivate_router_interface(self, params):
        """
        deactivate_router_interface: Deactivate a router
  interface
        API reference link: https://www.alibabacloud.
com/help/doc-detail/36033.htm
        """
        try:
            request = DeactivateRouterInterfaceReque st.
DeactivateRouterInterfaceReque st()
            # The ID of the router interface
            request.set_RouterInterfaceId(params['
router_interface_id'])
            response = self.client.do_action_with_excep
tion(request)
            response_json = json.loads(response)
            # Check if the router interface is in the
  available state
            if CheckStatus.check_status(TIME_DEFAU
LT_OUT * 5, DEFAULT_TIME * 5,
                                        self.describe_r i_status
,
                                        'Inactive', params['
router_interface_id']):
                return response_json
        except ServerExce ption as e:
            ExceptionH andler.server_exc eption(e)
        except ClientExce ption as e:
            ExceptionH andler.client_exc eption(e)

    def modify_rou ter_interf ace_attrib ute(self, params
):
        """
        modify_rou ter_interf ace_attrib ute: Modify the
configurat ions of a router interface
        API reference link: https://www.alibabaclo ud.
com/help/doc-detail/36036.htm
        """
        try:
            request = ModifyRout erInterfac eAttribute Request
.ModifyRout erInterfac eAttribute Request()
            # The ID of the router interface
            request.set_RouterInterfaceId(params['
router_interface_id'])
            # The ID of the peer router interface
            request.set_Opposi teInterfac eId(params['
opposite_interface_id'])
            # The ID of the peer VRouter
            request.set_Opposi teRouterId(params['
opposite_router_id'])
            response = self.client.do_action_with_excep
tion(request)
            response_json = json.loads(response)
            # Check if the router interface is in the
  available state
            if CheckStatus.check_status(TIME_DEFAU
LT_OUT, DEFAULT_TIME,
                                        self.describe_r i_status
,
                                        'Idle', params['
router_interface_id']):
                return response_json
        except ServerExce ption as e:
            ExceptionH andler.server_exc eption(e)
```

```
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )



    def   describe_r  outer_inte  rface ( self ,  instance_i  d ):
        """
        describe_r  outer_inte  rface :  Query   router
interfaces   in   a   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36035 . htm
        """
        try :
            request  =  DescribeRo  uterInterf  acesReques  t .
DescribeRo  uterInterf  acesReques  t ()
           #  The   queried   filter   type
            request . add_query_  param (' Filter . 1 . Key ', "
RouterInte  rfaceId ")
           #  The   queried   instance   ID
            request . add_query_  param (' Filter . 1 . Value . 1
',  instance_i  d )
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_r  i_status ( self ,  instance_i  d ):
        """
        describe_r  i_status :  Query   the   status   of   router
  interfaces   in   a   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36035 . htm
        """
        response  =  self . describe_r  outer_inte  rface (
instance_i  d )
        if   len ( response [' RouterInte  rfaceSet ']['
RouterInte  rfaceType ']) ==  0 :
            return  ''
        return   response [' RouterInte  rfaceSet '][' RouterInte
rfaceType '][ 0 ][' Status ']

    def   delete_rou  ter_interf  ace ( self ,  params ):
        """
        delete_rou  ter_interf  ace :  Delete   a   router
interface
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36034 . htm
        """
        try :
            request  =  DeleteRout  erInterfac  eRequest .
DeleteRout  erInterfac  eRequest ()
           #  The   ID   of   the   router   interface
            request . set_Router  InterfaceI  d ( params ['
instance_i  d '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
           #  Check   if   the   router   interface   is   in   the
  available   state
```

```
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME  *  5 ,
                                        self . describe_r  i_status
,
                                       '',  params [' instance_i  d
']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

def   main ():
    client  =  ACS_CLIENT
    router_int  erface  =  RouterInte  rface ( client )
    route_entr  y  =  RouteEntry ( client )

    params  = {}
    params [' spec '] = " Large . 2 "
    params [' role '] = " Initiating  Side "
    params [' router_id '] =  ROUTER_ID
    params [' router_typ  e '] = " VRouter "
    params [' opposite_r  egion_id '] = " cn - hangzhou "
    params [' opposite_r  outer_type '] = " VRouter "

   #  Create   the   initiator   router   interface
    router_int  erface_jso  n  =  router_int  erface . create_rou
ter_interf  ace ( params )
    CommonUtil . log (" create_rou  ter_interf  ace ",  router_int
erface_jso  n )

   #  Create   the   acceptor   router   interface
    params [' spec '] = " Negative "
    params [' role '] = " AcceptingS  ide "
    params [' router_id '] =  ROUTER_ID2
    router_int  erface_jso  n2  =  router_int  erface . create_rou
ter_interf  ace ( params )
    CommonUtil . log (" create_rou  ter_interf  ace ",  router_int
erface_jso  n2 )

   #  Modify   the   initiator   router   interface
    params [' router_int  erface_id '] =  router_int  erface_jso  n
[' RouterInte  rfaceId ']
    params [' opposite_i  nterface_i  d '] =  router_int
erface_jso  n2 [' RouterInte  rfaceId ']
    params [' opposite_r  outer_id '] =  ROUTER_ID2
    modify_ri_  json  =  router_int  erface . modify_rou
ter_interf  ace_attrib  ute ( params )
    CommonUtil . log (" modify_rou  ter_interf  ace_attrib  ute ",
modify_ri_  json )

   #  Modify   the   acceptor   router   interface
    params [' router_int  erface_id '] =  router_int  erface_jso
n2 [' RouterInte  rfaceId ']
    params [' opposite_i  nterface_i  d '] =  router_int
erface_jso  n [' RouterInte  rfaceId ']
    params [' opposite_r  outer_id '] =  ROUTER_ID
    modify_ri_  json2  =  router_int  erface . modify_rou
ter_interf  ace_attrib  ute ( params )
    CommonUtil . log (" modify_rou  ter_interf  ace_attrib  ute ",
modify_ri_  json2 )

   #  Query   the   initiator   router   interface
```

```
    describe_r  i_json  =  router_int  erface . describe_r
outer_inte  rface ( router_int  erface_jso  n [' RouterInte
rfaceId '])
    CommonUtil . log (" describe_r  outer_inte  rface ",
describe_r  i_json )

   #  Query   the   acceptor   router   interface
    describe_r  i_json2  =  router_int  erface . describe_r
outer_inte  rface ( router_int  erface_jso  n2 [' RouterInte
rfaceId '])
    CommonUtil . log (" describe_r  outer_inte  rface ",
describe_r  i_json2 )

   #  Initiate   the   connection
    params [' router_int  erface_id '] =  router_int  erface_jso  n
[' RouterInte  rfaceId ']
    connect_ri  _json  =  router_int  erface . connect_ro
uter_inter  face ( params )
    CommonUtil . log (" connect_ro  uter_inter  face ",
connect_ri  _json )

   #  Create   a   route   entry   whose   next   hop   is   the
initiator   router   interface
    params [' route_tabl  e_id '] =  TABLE_ID
    params [' destinatio  n_cidr_blo  ck '] = " 0 . 0 . 0 . 0 / 0 "
    params [' nexthop_ty  pe '] = ' RouterInte  rface '
    params [' nexthop_id '] =  router_int  erface_jso  n ['
RouterInte  rfaceId ']
    route_entr  y_json  =  route_entr  y . create_rou  te_entry (
params )
    CommonUtil . log (" create_rou  te_entry ",  route_entr  y_json
)

   #  Create   a   route   entry   whose   next   hop   is   the
acceptor   router   interface
    params [' route_tabl  e_id '] =  TABLE_ID2
    params [' destinatio  n_cidr_blo  ck '] = " 0 . 0 . 0 . 0 / 0 "
    params [' nexthop_ty  pe '] = ' RouterInte  rface '
    params [' nexthop_id '] =  router_int  erface_jso  n2 ['
RouterInte  rfaceId ']
    route_entr  y_json2  =  route_entr  y . create_rou  te_entry (
params )
    CommonUtil . log (" create_rou  te_entry ",  route_entr
y_json2 )

   #  Delete   the   route   entry   whose   next   hop   is   the
  acceptor   router   interface
    route_entr  y_json  =  route_entr  y . delete_rou  te_entry (
params )
    CommonUtil . log (" delete_rou  te_entry ",  route_entr  y_json
)

   #  Delete   the   route   entry   whose   next   hop   is   the
  initiator   router   interface
    params [' route_tabl  e_id '] =  TABLE_ID
    params [' nexthop_id '] =  router_int  erface_jso  n ['
RouterInte  rfaceId ']
    route_entr  y_json  =  route_entr  y . delete_rou  te_entry (
params )
    CommonUtil . log (" delete_rou  te_entry ",  route_entr  y_json
)

   #  Deactivate   the   initiator   router   interface
```

```
    params [' router_int  erface_id '] =  router_int  erface_jso  n
[' RouterInte  rfaceId ']
    deactivate  _ri_json  =  router_int  erface . deactivate
_router_in  terface ( params )
    CommonUtil . log (" deactivate  _router_in  terface ",
deactivate  _ri_json )

   #  Deactivate   the   acceptor   router   interface
    params [' router_int  erface_id '] =  router_int  erface_jso
n2 [' RouterInte  rfaceId ']
    deactivate  _ri_json2  =  router_int  erface . deactivate
_router_in  terface ( params )
    CommonUtil . log (" deactivate  _router_in  terface ",
deactivate  _ri_json2 )

   #  Delete   the   initiator   router   interface
    params [' instance_i  d '] =  router_int  erface_jso  n ['
RouterInte  rfaceId ']
    router_int  erface_jso  n  =  router_int  erface . delete_rou
ter_interf  ace ( params )
    CommonUtil . log (" delete_rou  ter_interf  ace ",  router_int
erface_jso  n )

   #  Delete   the   acceptor   router   interface
    params [' instance_i  d '] =  router_int  erface_jso  n2 ['
RouterInte  rfaceId ']
    router_int  erface_jso  n2  =  router_int  erface . delete_rou
ter_interf  ace ( params )
    CommonUtil . log (" delete_rou  ter_interf  ace ",  router_int
erface_jso  n2 )


if   __name__  == ' __main__ ':
    sys . exit ( main ())
```

5. **Access the** *ec_same_ac count . py* **directory and run the following command to interconnect two VPCs that are located in the same region and are under the same account.**

```
python   ec_same_ac  count . py
```

The system displays the following output:

```
--------------------------- create_rou  ter_interf  ace
 ---------------------------
{
  " RequestId ": " F5493DC1 - 53B0 - 4916 - 874F - 87773A5452  5F
 ",
  " RouterInte  rfaceId ": " ri - bp1ujwb6xs  w16xxxxxxx  x "
}
--------------------------- create_rou  ter_interf  ace
 ---------------------------
{
  " RequestId ": " E3F5BE99 - B2C5 - 4751 - 8173 - 0F590300EE  72
 ",
  " RouterInte  rfaceId ": " ri - bp1vze2rus  g2cxxxxxxx  x "
}
--------------------------- modify_rou  ter_interf  ace_attrib  ute
 ---------------------------
```

```
{
  " RequestId ": " 8D691507 - F31A - 41E5 - 9C72 - 457EFF1F77  27 "
}
--------------------------- modify_rou  ter_interf  ace_attrib  ute
 ---------------------------
{
  " RequestId ": " 2E664208 - 8F37 - 4F19 - B719 - 00B4F1AF03  B2 "
}
--------------------------- describe_r  outer_inte  rface
 ---------------------------
{
  " TotalCount ":  1 ,
  " RouterInte  rfaceSet ": {
    " RouterInte  rfaceType ": [
      {
        " BusinessSt  atus ": " Normal ",
        " CreationTi  me ": " 2019 - 04 - 30T06 : 09 : 16Z ",
        " Role ": " Initiating  Side ",
        " OppositeRo  uterId ": " vrt - bp141no9pd  s2bxxxxxxx  x
 ",
        " Spec ": " Large . 2 ",
        " Status ": " Idle ",
        " EndTime ": " 2999 - 09 - 08T16 : 00 : 00Z ",
        " OppositeIn  terfaceSpe  c ": " Negative ",
        " RouterInte  rfaceId ": " ri - bp1ujwb6xs  w16xxxxxxx  x
 ",
        " RouterType ": " VRouter ",
        " OppositeBa  ndwidth ":  0 ,
        " OppositeVp  cInstanceI  d ": " vpc - bp1v31by9j
 ix2xxxxxxx  x ",
        " HasReserva  tionData ":  false ,
        " OppositeIn  terfaceBus  inessStatu  s ": " Normal ",
        " OppositeRo  uterType ": " VRouter ",
        " OppositeRe  gionId ": " cn - hangzhou ",
        " VpcInstanc  eId ": " vpc - bp15opprpg  0rgxxxxxxx  x ",
        " RouterId ": " vrt - bp1ltkytn6  lgmxxxxxxx  x ",
        " CrossBorde  r ":  false ,
        " OppositeIn  terfaceOwn  erId ": "",
        " Bandwidth ":  2048 ,
        " OppositeIn  terfaceId ": " ri - bp1vze2rus  g2cxxxxxxx  x
 ",
        " ChargeType ": " AfterPay "
      }
    ]
  },
  " PageNumber ":  1 ,
  " RequestId ": " 89EF0631 - 0A36 - 41AD - A586 - AF4FFDA6E6  8B
 ",
  " PageSize ":  10
}
--------------------------- describe_r  outer_inte  rface
 ---------------------------
{
  " TotalCount ":  1 ,
  " RouterInte  rfaceSet ": {
    " RouterInte  rfaceType ": [
      {
        " Status ": " Idle ",
        " OppositeRe  gionId ": " cn - hangzhou ",
        " BusinessSt  atus ": " Normal ",
        " OppositeRo  uterId ": " vrt - bp1ltkytn6  lgmxxxxxxx  x
 ",
        " VpcInstanc  eId ": " vpc - bp1v31by9j  ix2xxxxxxx  x ",
```

```
        " RouterInte  rfaceId ": " ri - bp1vze2rus  g2cxxxxxxx  x
 ",
        " CreationTi  me ": " 2019 - 04 - 30T06 : 09 : 18Z ",
        " RouterType ": " VRouter ",
        " OppositeIn  terfaceOwn  erId ": "",
        " RouterId ": " vrt - bp141no9pd  s2bxxxxxxx  x ",
        " Bandwidth ":  0 ,
        " OppositeIn  terfaceId ": " ri - bp1ujwb6xs  w16xxxxxxx  x
 ",
        " EndTime ": " 2999 - 09 - 08T16 : 00 : 00Z ",
        " ChargeType ": " AfterPay ",
        " OppositeVp  cInstanceI  d ": " vpc - bp15opprpg
 0rgxxxxxxx  x ",
        " HasReserva  tionData ":  false ,
        " CrossBorde  r ":  false ,
        " OppositeIn  terfaceBus  inessStatu  s ": " Normal ",
        " Spec ": " Negative ",
        " OppositeRo  uterType ": " VRouter ",
        " Role ": " AcceptingS  ide "
      }
    ]
  },
  " PageNumber ":  1 ,
  " RequestId ": " 578448D7 - 9DCF - 4703 - 8337 - EF88DDF2C3  25
 ",
  " PageSize ":  10
}
--------------------------- connect_ro  uter_inter  face
 ---------------------------
{
  " RequestId ": " A5DBB86B - F6F5 - 4A53 - 899E - 8CF4FEF510  F2 "
}
--------------------------- create_rou  te_entry
 ---------------------------
{
  " RequestId ": " 70D896FE - 986B - 48EF - 9734 - 17D6BDC832  7A "
}
--------------------------- create_rou  te_entry
 ---------------------------
{
  " RequestId ": " A2233E25 - 4D6B - 4713 - A96F - E7CA745973  CA "
}
--------------------------- delete_rou  te_entry
 ---------------------------
{
  " RequestId ": " 464C62A4 - EE65 - 4414 - AF0A - 4984AE6B86  96 "
}
--------------------------- delete_rou  te_entry
 ---------------------------
{
  " RequestId ": " 0C11A332 - 969B - 47CA - A683 - 5BFFECA28B  3D "
}
--------------------------- deactivate  _router_in  terface
 ---------------------------
{
  " RequestId ": " 018305AD - FB9E - 450A - 91E8 - 3830634F5A  C2 "
}
--------------------------- deactivate  _router_in  terface
 ---------------------------
{
  " RequestId ": " 89B03203 - 9224 - 4CA3 - 8679 - E0A49029A2  D2 "
}
--------------------------- delete_rou  ter_interf  ace
 ---------------------------
```

```
{
  " RequestId ": " FB0424CE - D0C7 - 438B - A3FA - BCF24EE9CC  8A "
}
--------------------------- delete_rou  ter_interf  ace
 ---------------------------
{
  " RequestId ": " 8A0A0BB6 - A69D - 461B - A117 - 14AD6670DE  CA "
}
```

2.6.2 Interconnect a VBR and a VPC in the same region and under the same account

This topic describes how to use the Alibaba Cloud Python SDK to interconnect a VBR and a VPC that are located in the same region and are under the same account.

Prerequisites

Before interconnect a VBR and a VPC under the same account and in the same region, you must ensure that:

- A physical connection interface is applied.

> **Note:**
> Your physical connection interface and VPC must be under the same account and in the same region.

Context

Note the following when you create router interfaces:

- Only one pair of interconnected router interfaces can be created between two routers.
- Up to five router interfaces can be created on a router.
- If a router interface with an overdue payment is under your account, you cannot create router interfaces. To resolve this issue, add funds to your account.
- Route entries in the same route table cannot have the same destination CIDR blocks.
- The VBR must operate as the connection initiator and be in the Activated state.

The detailed steps used in this tutorial to interconnect a VBR and a VPC located in the same region and under the same account are as follows:

1. Create a VBR.
2. Create the initiator router interface.
3. Create the acceptor router interface.

4. Modify the initiator router interface.
5. Modify the acceptor router interface.
6. Query the initiator router interface.
7. Query the acceptor router interface.
8. Query the ID of the route table in the VBR.
9. Connect the initiator router interface to the acceptor router interface.
10. Create a route entry whose next hop is the initiator router interface.
11. Create a route entry whose next hop is the acceptor router interfac
12. Delete the route entry whose next hop is the acceptor router interface.
13. Delete the route entry whose next hop is the initiator router interface.
14. Deactivate the initiator router interface.
15. Deactivate the acceptor router interface.
16. Delete the initiator router interface.
17. Delete the acceptor router interface.
18. Delete the VBR.

Procedure

1. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ sdk_lib* folder.
2. Open the *consts . py* file in your text editor and configure the `ACS_CLIENT` parameter used for user authentication.
3. In the downloaded SDK directory, open the *$ aliyun - openapi - python - sdk - examples \ sdk_exampl es \ examples \ ec* folder.
4. Open the *ec_vbr_vpc . py* file in your text editor and configure your required parameters. Then, save the configurations and exit the text editor.

```
# encoding = utf - 8
 import   sys
 import   json
 from   aliyunsdkc  ore . acs_except  ion . exceptions   import
 ServerExce  ption ,  ClientExce  ption
 from   aliyunsdkv  pc . request . v20160428   import   CreateRout
 erInterfac  eRequest
 from   aliyunsdkv  pc . request . v20160428   import   DeleteRout
 erInterfac  eRequest
 from   aliyunsdkv  pc . request . v20160428   import   DescribeRo
 uterInterf  acesReques  t
 from   aliyunsdkv  pc . request . v20160428   import   ConnectRou
 terInterfa  ceRequest
 from   aliyunsdkv  pc . request . v20160428   import   Deactivate
 RouterInte  rfaceReque  st
```

```
from aliyunsdkvpc.request.v20160428 import ModifyRouterInterfaceAttributeRequest
from aliyunsdkvpc.request.v20160428 import CreateVirtualBorderRouterRequest
from aliyunsdkvpc.request.v20160428 import DescribeVirtualBorderRoutersRequest
from aliyunsdkvpc.request.v20160428 import DescribeRouteTablesRequest
from aliyunsdkvpc.request.v20160428 import DeleteVirtualBorderRouterRequest
from sdk_lib.sdk_route_entry import RouteEntry
from sdk_lib.exception import ExceptionHandler
from sdk_lib.common_util import CommonUtil
from sdk_lib.check_status import CheckStatus
from sdk_lib.consts import *

client = ACS_CLIENT

class RouterInterface(object):
    def __init__(self, client):
        self.client = client

    def create_router_interface(self, params):
        """
        create_router_interface: Create a router interface
        API reference link: https://www.alibabacloud.com/help/doc-detail/36032.htm
        """
        try:
            request = CreateRouterInterfaceRequest.CreateRouterInterfaceRequest()
            # The specification of the router interface
            request.set_Spec(params['spec'])
            # The role of the router interface
            request.set_Role(params['role'])
            # The ID of the router associated with the router interface
            request.set_RouterId(params['router_id'])
            # The type of the router associated with the router interface
            request.set_RouterType(params['router_type'])
            # The ID of the region to which the connection acceptor belongs
            request.set_OppositeRegionId(params['opposite_region_id'])
            # The type of the router associated with the peer router interface
            request.set_OppositeRouterType(params['opposite_router_type'])
            # The ID of the access point to which the peer end belongs
            request.set_OppositeAccessPointId(params['opposite_access_point_id'])
            # The ID of the access point to which the VBR belongs
            # request.set_AccessPointId(params['access_point_id'])

            response = self.client.do_action_with_exception(request)
            response_json = json.loads(response)
```

```
            #  Check   if   the   router   interface   is   in   the
  available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                          self . describe_r  i_status
,
                                         ' Idle ',  response_j  son
[' RouterInte  rfaceId ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   create_rou  ter_interf  ace_vbr ( self ,  params ):
        """
        create_rou  ter_interf  ace :  Create   a   router
interface
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36032 . htm
        """
        try :
            request  =  CreateRout  erInterfac  eRequest .
CreateRout  erInterfac  eRequest ()
            #  The   specificat  ion   of   the   router
interface
            request . set_Spec ( params [' spec '])
            #  The   role   of   the   router   interface
            request . set_Role ( params [' role '])
            #  The   ID   of   the   router   associated   with
the   router   interface
            request . set_Router  Id ( params [' router_id '])
            #  The   type   of   the   router   associated   with
the   router   interface
            request . set_Router  Type ( params [' router_typ  e
'])
            #  The   ID   of   the   region   to   which   the
connection   acceptor   belongs
            request . set_Opposi  teRegionId ( params ['
opposite_r  egion_id '])
            #  The   type   of   the   router   associated   with
the   peer   router   interface
            request . set_Opposi  teRouterTy  pe ( params ['
opposite_r  outer_type '])
            #  The   ID   of   the   access   point   to   which
the   VBR   belongs
            request . set_Access  PointId ( params [' access_poi
nt_id '])

            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            #  Check   if   the   router   interface   is   in   the
  available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                          self . describe_r  i_status
,
                                         ' Idle ',  response_j  son
[' RouterInte  rfaceId ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
```

```
            ExceptionH  andler . client_exc  eption ( e )

    def   connect_ro  uter_inter  face ( self ,  params ):
        """
        connect_ro  uter_inter  face :  The   initiator   router
interface   initiates   the   connection   to   the   acceptor
router   interface
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36031 . htm
        """
        try :
            request  =  ConnectRou  terInterfa  ceRequest .
ConnectRou  terInterfa  ceRequest ()
           #  The   ID   of   the   initiator   router   interface
            request . set_Router  InterfaceI  d ( params ['
router_int  erface_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT  *  5 ,  DEFAULT_TI  ME  *  5 ,
                                        self . describe_r  i_status
,
                                       ' Active ',  params ['
router_int  erface_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   deactivate  _router_in  terface ( self ,  params ):
        """
        deactivate  _router_in  terface :  Deactivate   a   router
  interface
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36033 . htm
        """
        try :
            request  =  Deactivate  RouterInte  rfaceReque  st .
Deactivate  RouterInte  rfaceReque  st ()
           #  The   ID   of   the   router   interface
            request . set_Router  InterfaceI  d ( params ['
router_int  erface_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
           #  Check   if   the   router   interface   is   in   the
  available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT  *  5 ,  DEFAULT_TI  ME  *  5 ,
                                        self . describe_r  i_status
,
                                       ' Inactive ',  params ['
router_int  erface_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   modify_rou  ter_interf  ace_attrib  ute ( self ,  params
):
        """
```

```
        modify_rou  ter_interf  ace_attrib  ute :  Modify   the
configurat  ions   of   a   router   interface
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36036 . htm
       """
        try :
            request  =  ModifyRout  erInterfac  eAttribute  Request
. ModifyRout  erInterfac  eAttribute  Request ()
           #  The   ID   of   the   router   interface
            request . set_Router  InterfaceI  d ( params ['
router_int  erface_id '])
           #  The   ID   of   the   peer   router   interface
            request . set_Opposi  teInterfac  eId ( params ['
opposite_i  nterface_i  d '])
           #  The   ID   of   the   peer   router
            request . set_Opposi  teRouterId ( params ['
opposite_r  outer_id '])
           #  The   type   of   the   peer   router
            request . set_Opposi  teRouterTy  pe ( params ['
opposite_r  outer_type '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
           #  Check   if   the   router   interface   is   in   the
  available   state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                       self . describe_r  i_status
,
                                      ' Idle ',  params ['
router_int  erface_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )



    def   describe_r  outer_inte  rface ( self ,  instance_i  d ):
       """
        describe_r  outer_inte  rface : Query   router   interfaces
  in   a   region
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36035 . htm
       """
        try :
            request  =  DescribeRo  uterInterf  acesReques  t .
DescribeRo  uterInterf  acesReques  t ()
           #  The   queried   filter   type
            request . add_query_  param (' Filter . 1 . Key ', "
RouterInte  rfaceId ")
           #  The   queried   instance   ID
            request . add_query_  param (' Filter . 1 . Value . 1
',  instance_i  d )
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )
```

```
    def describe_r i_status ( self , instance_i d ):
        """
        describe_r i_status : Query the status of router
  interfaces in a region
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36035 . htm
        """
        response = self . describe_r outer_inte rface (
instance_i d )
        if len ( response [' RouterInte rfaceSet ']['
RouterInte rfaceType ']) == 0 :
            return ''
        return response [' RouterInte rfaceSet '][' RouterInte
rfaceType '][ 0 ][' Status ']

    def delete_rou ter_interf ace ( self , params ):
        """
        delete_rou ter_interf ace : Delete a router
interface
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36034 . htm
        """
        try :
            request = DeleteRout erInterfac eRequest .
DeleteRout erInterfac eRequest ()
           # The ID of the router interface
            request . set_Router InterfaceI d ( params ['
instance_i d '])
            response = self . client . do_action_ with_excep
tion ( request )
            response_j son = json . loads ( response )
           # Check if the router interface is in the
  available state
            if CheckStatu s . check_stat us ( TIME_DEFAU
LT_OUT , DEFAULT_TI ME * 5 ,
                                        self . describe_r i_status
,
                                       '', params [' instance_i d
']):
                return response_j son
        except ServerExce ption as e :
            ExceptionH andler . server_exc eption ( e )
        except ClientExce ption as e :
            ExceptionH andler . client_exc eption ( e )

    def create_vir tual_borde r_router ( self , params ):
        """
        create_vir tual_borde r_router : Create a VBR
        API reference link : https :// www . alibabaclo ud .
com / help / doc - detail / 36023 . htm
        """
        try :
            request = CreateVirt ualBorderR outerReque st .
CreateVirt ualBorderR outerReque st ()
           # The ID of the physical connection
            request . set_Physic alConnecti onId ( params ['
physical_c onnection_ id '])
           # The Alibaba Cloud - side IP address of
the VBR
            request . set_LocalG atewayIp ( params [' local_gate
way_ip '])
           # The peer IP address of the physical
connection - side interface of the VBR
```

```
            request . set_PeerGa  tewayIp ( params [' peer_gatew
ay_ip '])
           #  The   subnet   mask   for   the   Alibaba   Cloud -
side   and   customer - side   IP   addresses   of   the   VBR
            request . set_Peerin  gSubnetMas  k ( params ['
peering_su  bnet_mask '])
           #  The   VLAN   ID   of   the   VBR
            request . set_VlanId ( params [' vlan_id '])

            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
           #  Check   if   the   VBR   is   in   the   available
state
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                          self . describe_v
br_status ,
                                         ' active ',  response_j  son
[' VbrId ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   delete_vir  tual_borde  r_router ( self ,  params ):
        """
        delete_vir  tual_borde  r_router :  Delete   the   VBR
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 40542 . htm
        """
        try :
            request  =  DeleteVirt  ualBorderR  outerReque  st .
DeleteVirt  ualBorderR  outerReque  st ()
            request . set_VbrId ( params [' vbr_id '])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            if   CheckStatu  s . check_stat  us ( TIME_DEFAU
LT_OUT ,  DEFAULT_TI  ME ,
                                          self . describe_v
br_status ,
                                         '',  params [' vbr_id ']):
                return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_v  irtual_bor  der_router ( self ,  instance_i
d ):
        """
        describe_v  irtual_bor  der_router :  Query   VBRs
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36024 . htm
        """
        try :
            request  =  DescribeVi  rtualBorde  rRoutersRe  quest .
DescribeVi  rtualBorde  rRoutersRe  quest ()
           #  The   queried   filter   type
            request . add_query_  param (' Filter . 1 . Key ', "
VbrId ")
           #  The   queried   instance   ID
```

```
            request . add_query_  param (' Filter . 1 . Value . 1
',  instance_i  d )
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )

    def   describe_v  br_status ( self ,  instance_i  d ):
        """
        describe_v  irtual_bor  der_router :  Query   the   status
  of   VBRs
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36024 . htm
        """
        response  =  self . describe_v  irtual_bor  der_router (
instance_i  d )
        if   len ( response [' VirtualBor  derRouterS  et '][']
VirtualBor  derRouterT  ype ']) ==  0 :
            return  ''
        return   response [' VirtualBor  derRouterS  et '][']
VirtualBor  derRouterT  ype '][ 0 ][' Status ']

    def   describe_r  oute_table ( self ,  params ):
        """
        describe_r  oute_table :  Query   route   tables
        API   reference   link :  https :// www . alibabaclo  ud .
com / help / doc - detail / 36014 . htm
        """
        try :
            request  =  DescribeRo  uteTablesR  equest .
DescribeRo  uteTablesR  equest ()
            #  The   ID   of   the   VRouter   or   VBR   to   which
  the   route   table   belongs
            request . set_Router  Id ( params [' router_id '])
            #  The   type   of   the   router   to   which   the
route   table   belongs
            request . set_Router  Type ( params [' router_typ  e
'])
            response  =  self . client . do_action_  with_excep
tion ( request )
            response_j  son  =  json . loads ( response )
            return   response_j  son
        except   ServerExce  ption   as   e :
            ExceptionH  andler . server_exc  eption ( e )
        except   ClientExce  ption   as   e :
            ExceptionH  andler . client_exc  eption ( e )


def   main ():
    client  =  ACS_CLIENT
    router_int  erface  =  RouterInte  rface ( client )
    route_entr  y  =  RouteEntry ( client )

    params  = {}
    params [' spec '] = " Large . 2 "
    params [' role '] = " Initiating  Side "
    params [' router_typ  e '] = " VBR "
    params [' opposite_r  egion_id '] = " cn - hangzhou "
    params [' opposite_r  outer_type '] = " VRouter "
```

```
    params [' access_poi  nt_id '] = " ap - cn - hangzhou - xx - x
"

   #  Create   a   VBR
    params [' physical_c  onnection_  id '] =  PC_ID
    params [' local_gate  way_ip '] = " 116 . xx . xx . 254 "
    params [' peer_gatew  ay_ip '] = " 116 . xx . xx . 254 "
    params [' peering_su  bnet_mask '] = " 255 . 255 . 255 . 252 "
    params [' vlan_id '] =  30
    vbr_json  =  router_int  erface . create_vir  tual_borde
r_router ( params )
    CommonUtil . log (" create_vir  tual_borde  r_router ",
vbr_json )

   #  Create   an   initiator   router   interface
    params [' router_id '] =  vbr_json [' VbrId ']
    router_int  erface_jso  n  =  router_int  erface . create_rou
ter_interf  ace_vbr ( params )
    CommonUtil . log (" create_rou  ter_interf  ace ",  router_int
erface_jso  n )

   #  Create   an   acceptor   router   interface
    params [' router_typ  e '] = " VRouter "
    params [' spec '] = " Negative "
    params [' role '] = " AcceptingS  ide "
    params [' router_id '] =  ROUTER_ID2
    params [' opposite_r  outer_type '] = ' VBR '
    params [' opposite_a  ccess_poin  t_id '] =  params ['
access_poi  nt_id ']
    router_int  erface_jso  n2  =  router_int  erface . create_rou
ter_interf  ace ( params )
    CommonUtil . log (" create_rou  ter_interf  ace ",  router_int
erface_jso  n2 )

   #  Modify   the   initiator   router   interface
    params [' router_int  erface_id '] =  router_int  erface_jso  n
[' RouterInte  rfaceId ']
    params [' opposite_i  nterface_i  d '] =  router_int
erface_jso  n2 [' RouterInte  rfaceId ']
    params [' opposite_r  outer_id '] =  ROUTER_ID2
    params [' opposite_r  outer_type '] = " VRouter "
    modify_ri_  json  =  router_int  erface . modify_rou
ter_interf  ace_attrib  ute ( params )
    CommonUtil . log (" modify_rou  ter_interf  ace_attrib  ute ",
modify_ri_  json )

   #  Modify   the   acceptor   router   interface
    params [' router_int  erface_id '] =  router_int  erface_jso
n2 [' RouterInte  rfaceId ']
    params [' opposite_i  nterface_i  d '] =  router_int
erface_jso  n [' RouterInte  rfaceId ']
    params [' opposite_r  outer_id '] =  vbr_json [' VbrId ']
    params [' opposite_r  outer_type '] = " VBR "
    modify_ri_  json2  =  router_int  erface . modify_rou
ter_interf  ace_attrib  ute ( params )
    CommonUtil . log (" modify_rou  ter_interf  ace_attrib  ute ",
modify_ri_  json2 )

   #  Query   the   initiator   router   interface
    describe_r  i_json  =  router_int  erface . describe_r
outer_inte  rface ( router_int  erface_jso  n [' RouterInte
rfaceId '])
    CommonUtil . log (" describe_r  outer_inte  rface ",
describe_r  i_json )
```

```
   #  Query   the   acceptor   router   interface
    describe_r  i_json2  =  router_int  erface . describe_r
outer_inte  rface ( router_int  erface_jso  n2 [' RouterInte
rfaceId '])
    CommonUtil . log (" describe_r  outer_inte  rface ",
describe_r  i_json2 )

   #  Query   the   ID   of   the   route   table   of   the   VBR
    params [' router_id '] =  vbr_json [' VbrId ']
    params [' router_typ  e '] = ' VBR '
    route_tabl  e_json  =  router_int  erface . describe_r
oute_table ( params )
    CommonUtil . log (" describe_r  oute_table ",  route_tabl
e_json )

   #  Initiate   the   connection
    params [' router_int  erface_id '] =  router_int  erface_jso  n
[' RouterInte  rfaceId ']
    connect_ri  _json  =  router_int  erface . connect_ro
uter_inter  face ( params )
    CommonUtil . log (" connect_ro  uter_inter  face ",
connect_ri  _json )

   #  Create   a   route   entry   whose   next   hop   is   the
initiator   router   interface
    params [' route_tabl  e_id '] =  route_tabl  e_json ["
RouteTable  s "][" RouteTable "][ 0 ][" RouteTable  Id "]
    params [' destinatio  n_cidr_blo  ck '] = " 0 . 0 . 0 . 0 / 0 "
    params [' nexthop_ty  pe '] = ' RouterInte  rface '
    params [' nexthop_id '] =  router_int  erface_jso  n ['
RouterInte  rfaceId ']
    route_entr  y_json  =  route_entr  y . create_rou  te_entry (
params )
    CommonUtil . log (" create_rou  te_entry ",  route_entr  y_json
)

   #  Create   a   route   entry   whose   next   hop   is   the
acceptor   router   interface
    params [' route_tabl  e_id '] =  TABLE_ID2
    params [' destinatio  n_cidr_blo  ck '] = " 0 . 0 . 0 . 0 / 0 "
    params [' nexthop_ty  pe '] = ' RouterInte  rface '
    params [' nexthop_id '] =  router_int  erface_jso  n2 ['
RouterInte  rfaceId ']
    route_entr  y_json2  =  route_entr  y . create_rou  te_entry (
params )
    CommonUtil . log (" create_rou  te_entry ",  route_entr
y_json2 )

   #  Delete   the   route   entry   whose   next   hop   is   the
  acceptor   router   interface
    route_entr  y_json  =  route_entr  y . delete_rou  te_entry (
params )
    CommonUtil . log (" delete_rou  te_entry ",  route_entr  y_json
)

   #  Delete   the   route   entry   whose   next   hop   is   the
  initiator   router   interface
    params [' route_tabl  e_id '] =  route_tabl  e_json ["
RouteTable  s "][" RouteTable "][ 0 ][" RouteTable  Id "]
    params [' nexthop_id '] =  router_int  erface_jso  n ['
RouterInte  rfaceId ']
    route_entr  y_json  =  route_entr  y . delete_rou  te_entry (
params )
```

```
    CommonUtil . log (" delete_rou  te_entry ",  route_entr  y_json
)

   #  Deactivate   the   initiator   router   interface
    params [' router_int  erface_id '] =  router_int  erface_jso  n
[' RouterInte  rfaceId ']
    deactivate  _ri_json  =  router_int  erface . deactivate
_router_in  terface ( params )
    CommonUtil . log (" deactivate  _router_in  terface ",
deactivate  _ri_json )

   #  Deactivate   the   acceptor   router   interface
    params [' router_int  erface_id '] =  router_int  erface_jso
n2 [' RouterInte  rfaceId ']
    deactivate  _ri_json2  =  router_int  erface . deactivate
_router_in  terface ( params )
    CommonUtil . log (" deactivate  _router_in  terface ",
deactivate  _ri_json2 )

   #  Delete   the   initiator   router   interface
    params [' instance_i  d '] =  router_int  erface_jso  n ['
RouterInte  rfaceId ']
    router_int  erface_jso  n  =  router_int  erface . delete_rou
ter_interf  ace ( params )
    CommonUtil . log (" delete_rou  ter_interf  ace ",  router_int
erface_jso  n )

   #  Delete   the   acceptor   router   interface
    params [' instance_i  d '] =  router_int  erface_jso  n2 ['
RouterInte  rfaceId ']
    router_int  erface_jso  n2  =  router_int  erface . delete_rou
ter_interf  ace ( params )
    CommonUtil . log (" delete_rou  ter_interf  ace ",  router_int
erface_jso  n2 )

   #  Delete   the   VBR
    params [' vbr_id '] =  vbr_json [' VbrId ']
    vbr_json  =  router_int  erface . delete_vir  tual_borde
r_router ( params )
    CommonUtil . log (" delete_vir  tual_borde  r_router ",
vbr_json )

if   __name__  == ' __main__ ':
    sys . exit ( main ())
```

5. **Access the** *ec_vbr_vpc . py* **directory and run the following command to interconnect a VBR and a VPC that are located in the same region and are under the same account.**

```
python   ec_vbr_vpc . py
```

The system displays the following output:

```
--------------------------- create_vir  tual_borde  r_router
 ---------------------------
{
  " VbrId ": " vbr - bp1vudncgk  9jtxxxxxxx  x ",
  " RequestId ": " 0C4FABDB - FF18 - 4E70 - 9E43 - 6DC03197F0  EA "
}
```

```
--------------------------- create_rou  ter_interf  ace
 ---------------------------
{
  " RequestId ": " 11F14FA2 - ECFA - 4B27 - A75D - 7C6D5FECAC  DF
 ",
  " RouterInte  rfaceId ": " ri - bp1vabte8n  bdexxxxxxx  x "
}
--------------------------- create_rou  ter_interf  ace
 ---------------------------
{
  " RequestId ": " 72FAB86D - 470B - 40E4 - 8476 - F9223A9114  86
 ",
  " RouterInte  rfaceId ": " ri - bp1mxkc61l  y0nxxxxxxx  x "
}
--------------------------- modify_rou  ter_interf  ace_attrib  ute
 ---------------------------
{
  " RequestId ": " 93D5FFF7 - 6C05 - 41B1 - 92F0 - AC8F1809BC  98 "
}
--------------------------- modify_rou  ter_interf  ace_attrib  ute
 ---------------------------
{
  " RequestId ": " 1C461452 - 42BD - 4649 - B318 - 2214222B4F  CA "
}
--------------------------- describe_r  outer_inte  rface
 ---------------------------
{
  " TotalCount ":  1 ,
  " RouterInte  rfaceSet ": {
    " RouterInte  rfaceType ": [
      {
        " BusinessSt  atus ": " Normal ",
        " CreationTi  me ": " 2019 - 04 - 30T02 : 54 : 22Z ",
        " AccessPoin  tId ": " ap - cn - hangzhou - xx - x ",
        " Role ": " Initiating  Side ",
        " OppositeRo  uterId ": " vrt - bp141no9pd  s2bxxxxxxx  x
 ",
        " Spec ": " Large . 2 ",
        " Status ": " Idle ",
        " EndTime ": " 2999 - 09 - 08T16 : 00 : 00Z ",
        " OppositeIn  terfaceSpe  c ": " Negative ",
        " RouterInte  rfaceId ": " ri - bp1vabte8n  bdexxxxxxx  x
 ",
        " RouterType ": " VBR ",
        " OppositeBa  ndwidth ":  0 ,
        " OppositeVp  cInstanceI  d ": " vpc - bp1v31by9j
 ix2xxxxxxx  x ",
        " HasReserva  tionData ":  false ,
        " OppositeIn  terfaceBus  inessStatu  s ": " Normal ",
        " OppositeRo  uterType ": " VRouter ",
        " OppositeRe  gionId ": " cn - hangzhou ",
        " RouterId ": " vbr - bp1vudncgk  9jtxxxxxxx  x ",
        " CrossBorde  r ":  false ,
        " OppositeIn  terfaceOwn  erId ": "",
        " Bandwidth ":  2048 ,
        " OppositeIn  terfaceId ": " ri - bp1mxkc61l  y0nxxxxxxx  x
 ",
        " ChargeType ": " AfterPay "
      }
    ]
  },
  " PageNumber ":  1 ,
  " RequestId ": " 3553DB38 - A4A0 - 4453 - 976C - 542E96B1B5  A9
 ",
```

```
  " PageSize ":  10
}
--------------------------- describe_r  outer_inte  rface
 ---------------------------
{
  " TotalCount ":  1 ,
  " RouterInte  rfaceSet ": {
    " RouterInte  rfaceType ": [
      {
        " Status ": " Idle ",
        " OppositeRe  gionId ": " cn - hangzhou ",
        " BusinessSt  atus ": " Normal ",
        " OppositeRo  uterId ": " vbr - bp1vudncgk  9jtxxxxxxx  x
 ",
        " VpcInstanc  eId ": " vpc - bp1v31by9j  ix2xxxxxxx  x ",
        " RouterInte  rfaceId ": " ri - bp1mxkc61l  y0nxxxxxxx  x
 ",
        " CreationTi  me ": " 2019 - 04 - 30T02 : 54 : 24Z ",
        " RouterType ": " VRouter ",
        " OppositeIn  terfaceOwn  erId ": "",
        " RouterId ": " vrt - bp141no9pd  s2bxxxxxxx  x ",
        " Bandwidth ":  0 ,
        " OppositeIn  terfaceId ": " ri - bp1vabte8n  bdexxxxxxx  x
 ",
        " EndTime ": " 2999 - 09 - 08T16 : 00 : 00Z ",
        " ChargeType ": " AfterPay ",
        " OppositeAc  cessPointI  d ": " ap - cn - hangzhou - xx - x
 ",
        " HasReserva  tionData ":  false ,
        " CrossBorde  r ":  false ,
        " OppositeIn  terfaceBus  inessStatu  s ": " Normal ",
        " Spec ": " Negative ",
        " OppositeRo  uterType ": " VBR ",
        " Role ": " AcceptingS  ide "
      }
    ]
  },
  " PageNumber ":  1 ,
  " RequestId ": " 217D8D94 - C508 - 4285 - 8101 - 7DE3B53A88  A5
 ",
  " PageSize ":  10
}
--------------------------- describe_r  oute_table
 ---------------------------
{
  " TotalCount ":  1 ,
  " PageNumber ":  1 ,
  " RequestId ": " CA6BBE52 - DF5E - 496A - 93CF - 9857AF22D2  AC
 ",
  " PageSize ":  10 ,
  " RouteTable  s ": {
    " RouteTable ": [
      {
        " RouteTable  Id ": " vtb - bp1s126yz0  swpxxxxxxx  x ",
        " RouteEntry  s ": {
          " RouteEntry ": []
        },
        " CreationTi  me ": " 2019 - 04 - 30T02 : 54 : 19Z ",
        " VSwitchIds ": {
          " VSwitchId ": []
        },
        " ResourceGr  oupId ": "",
        " VRouterId ": " vbr - bp1vudncgk  9jtxxxxxxx  x ",
        " RouteTable  Type ": " System "
```

```
      }
    ]
  }
}
--------------------------- connect_ro  uter_inter  face
 ---------------------------
{
  " RequestId ": " 93476D4E - 6C08 - 44B8 - 83E4 - 5759E1A08F  29 "
}
--------------------------- create_rou  te_entry
 ---------------------------
{
  " RequestId ": " 66D4F46B - 5E0F - 4565 - 8740 - 845EC26EBE  00 "
}
--------------------------- create_rou  te_entry
 ---------------------------
{
  " RequestId ": " E1F99FEF - 2499 - 40A7 - 84ED - 6F9D440D4F  F8 "
}
--------------------------- delete_rou  te_entry
 ---------------------------
{
  " RequestId ": " 8C983886 - F058 - 4234 - A6D7 - ECDEE2C2D9  45 "
}
--------------------------- delete_rou  te_entry
 ---------------------------
{
  " RequestId ": " DDEC56B9 - CDD4 - 4D0A - B460 - 040B85B97F  E9 "
}
--------------------------- deactivate  _router_in  terface
 ---------------------------
{
  " RequestId ": " 04069B7C - 6A42 - 43F2 - A086 - 50F8029400  45 "
}
--------------------------- deactivate  _router_in  terface
 ---------------------------
{
  " RequestId ": " B2EBF829 - E1C0 - 43E5 - 9BAD - DFFCBFA301  F7 "
}
--------------------------- delete_rou  ter_interf  ace
 ---------------------------
{
  " RequestId ": " 20EEC4A6 - E468 - 4F3C - A743 - 89193F7504  E1 "
}
--------------------------- delete_rou  ter_interf  ace
 ---------------------------
{
  " RequestId ": " 59DCE168 - B9CC - 43A3 - A54F - 0D8C194BAB  E0 "
}
--------------------------- delete_vir  tual_borde  r_router
 ---------------------------
{
  " RequestId ": " E1D71EEC - FE9F - 4834 - 8780 - 19EBBD91A6  38 "
}
```