ALIBABA CLOUD

Alibaba Cloud

Elastic Compute Service

SDK Sample

Document Version: 20200914

Alibaba Cloud

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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice 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.
Warning	A warning notice 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 restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.SDK reference

Currently, Alibaba Cloud ECS provides the following programming languages of SDK.

- Java
- Python
- PHP
- C++
- .NET

2.Java example

2.1. Install ECS SDK for Java

This topic describes how to install ECS SDK for Java and the Alibaba Cloud SDK core library.

Prerequisites

JDK 1.6 or later must be installed before you install ECS SDK for Java.

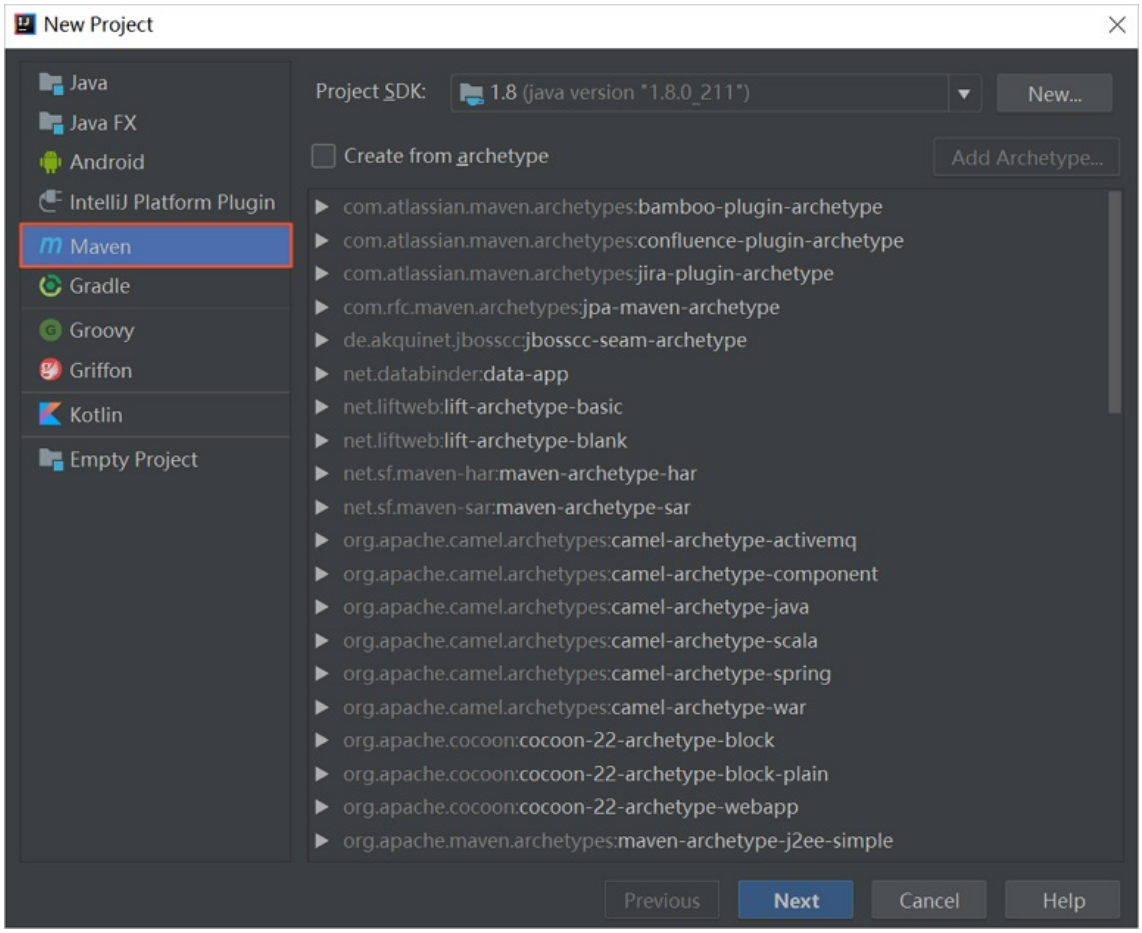
Context

The Alibaba Cloud Developer Center provides ECS SDK for Java, and Maven project dependencies and JAR packages of the Alibaba Cloud SDK core library. You can write code to call Alibaba Cloud SDKs to access Alibaba Cloud products and services.

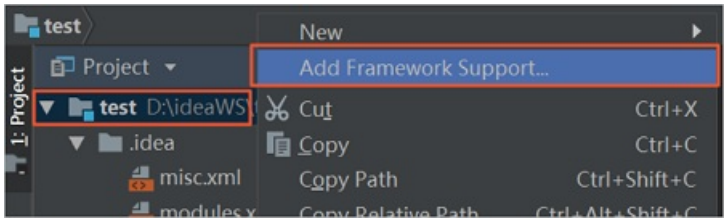
This example uses Windows 10 64-bit as the operating system and uses IntelliJ IDEA as the Java development tool.

Procedure

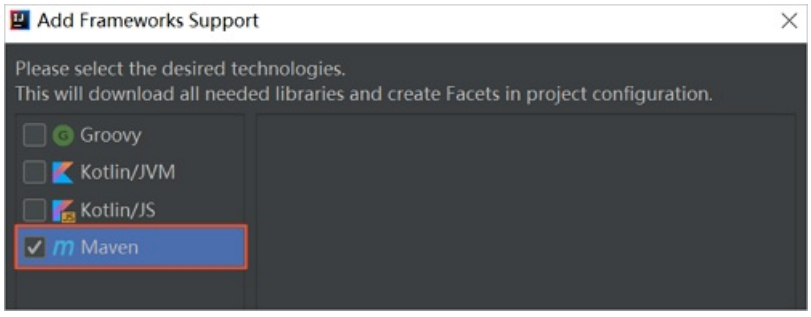
1. Choose one of the following ways to configure the Maven project management tool in IDEA:
   - Use the integrated Maven project management tool in IDEA.
   - Download the Maven software corresponding to the operating system from the official Maven website (Download Apache Maven) and manually configure the Maven tool.
2. Go to Alibaba Cloud SDKs to obtain the **Alibaba Cloud SDK core library** and Maven dependencies of **ECS**.
3. Create a Maven project by using one of the following methods:
   - Method 1: Create a Maven project in IDEA.



- Method 2: Convert an existing project to a Maven project.
  a. Right-click the project to be converted and choose **Add Framework Support...** from the shortcut menu.



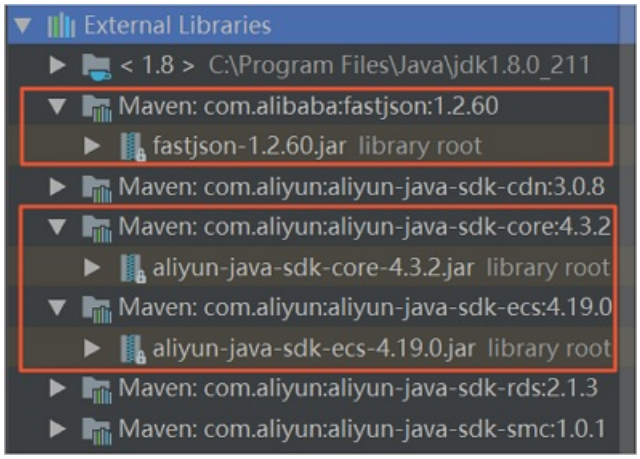
  b. Select **Maven** and click **OK**.



4. Add the aliyun-java-sdk-core, aliyun-java-sdk-ecs, and fastjson dependencies to the pom.xml file under the project directory. After the dependencies are added, the Maven project management tool automatically downloads the corresponding JAR packages.

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.4.3</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-ecs</artifactId>
    <version>4.17.1</version>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.60</version>
  </dependency>
</dependencies>
```

5. Check External Libraries under the project directory. If the output in the following figure is generated, the dependencies are imported.



2.2. General process

This topic describes how to use the ECS Java SDK by taking DescribeImages as an example. DescribeImages is the operation used by the ECS Java SDK to query available image resources.

Prerequisites

You have created an AccessKey pair. For information about how to create an AccessKey pair, see .

Note To protect the AccessKey pair of your Alibaba Cloud account, we recommend that you create a RAM user, grant the RAM user the permissions to access ECS instances, and then use the AccessKey pair of the RAM user to call the Java SDK. For more information, see Implement access control by using RAM.

Context

- In this example, the IClientProfile and IAcsClient classes are included in aliyun-java-sdk-core, and the other classes are included in aliyun-java-sdk-ecs.
- The purpose of this example is to query ECS public images. For information about public images, see image related documentation. For more information, see Overview.
- The following table compares the methods used by the previous SDK and by the new SDK, their classes, and objects. If you are using the previous SDK, we recommend that you switch to the new version to obtain the new features.

Item	New SDK	Earlier SDK
Submit a request	getAcsResponse()	execute()
Class that stores the AccessKey pair	IClientProfile	AliyunClient
Objects for storing identity credentials	DefaultProfile.getProfile(RegionId, AccessKey, AccessKeySecret)	new DefaultAliyunClient(APIUrl, AccessKey, AccessKeySecret)
Package name prefix	com.aliyuncs	com.aliyun.api

Procedure

1. Generate the profile object from the IClientProfile class.

   The profile object stores the region, AccessKey ID, and AccessKey secret, such as the cn-hangzhou in this example. For more information about regions, see .

   ```
   IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<yourAccessKeyID>", "<yourAccessKeySecret>");
   ```

2. Generate the object client of IAcsClient from the IClientProfile class.

   Subsequently obtain response from IClientProfile.

   ```
   IAcsClient client = new DefaultAcsClient(profile);
   ```

3. Create a request for the API opertion and use the constructor to generate a default class request.

   The class is named after the API operation name followed by Request. The API operation for obtaining the image list is named DescribeImages and the corresponding request class name is DescribeImagesRequest.

   ```
   DescribeImagesRequest request = new DescribeImagesRequest();
   ```

4. Set parameters for the request class.

   Set required parameters in the API operation through the setXxx method of the request class. Set parameters through the setXxx method. In the example:

   - The DescribeImages API operation uses RegionId to specify the region.
   - The DescribeImages API operation uses ImageOwnerAlias to specify the image type to be queried. The value of setImageOwnerAlias is system, which indicates public images.

   ```
   request.setImageOwnerAlias("system");
   ```

5. Obtain the response corresponding to the specified request parameter through theclient object.

   ```
   DescribeImagesResponse response = client.getAcsResponse(request);
   System.out.println(JSON.toJSONString(response));
   ```

6. Obtain the returned parameter value by calling the corresponding getXxx method in response.

   If you need to obtain the name of an image, you must obtain the collection of image objects by calling getImages(), the information of the target image by traversing the image objects, and the details of the image by calling getgetImageName() or getgetImageId.

   ```
   for(DescribeImagesResponse.Image image:response.getImages())
           {
              System.out.println(image.getImageId());
              System.out.println(image.getImageName());
           }
   ```

   Based on different API operations, the response may contain multi-layer information. For example, if you call the DescribeImages operation, the response is represented as a collection in the form of list that stores the information about each image. You must obtain the collection of image objects by calling getImages(), the information of an image by traversing the image objects, and the details of the image by calling getXxx.

7. Handle server and client errors by using **catch()**.

   - Server error

     ```
     catch (ServerException e) {
             e.printStackTrace();
     ```

   - Client error

     ```
     catch (ClientException e) {
             System.out.println("ErrCode:" + e.getErrCode());
             System.out.println("ErrMsg:" + e.getErrMsg());
             System.out.println("RequestId:" + e.getRequestId());
     ```

Result

- The complete response is as follows:


```
{
  "PageNumber": 1,
  "TotalCount": 43,
  "PageSize": 1,
  "RegionId": "cn-hangzhou",
  "RequestId": "C93F3D9F-CF25-47DF-9C0F-614395E5DCAC",
  "Images": {
    "Image": [
      {
        "ImageId": "freebsd_11_02_64_30G_alibase_20190722.vhd",
        "Description": "",
        "OSNameEn": "FreeBSD  11.2 64 bit",
        "ProductCode": "",
        "ResourceGroupId": "",
        "OSType": "linux",
        "Architecture": "x86_64",
        "OSName": "FreeBSD  11.2 64-bit",
        "DiskDeviceMappings": {
          "DiskDeviceMapping": []
        },
        "ImageOwnerAlias": "system",
        "Progress": "100%",
        "IsSupportCloudinit": false,
        "Usage": "instance",
        "CreationTime": "2019-07-23T05:41:06Z",
        "Tags": {
          "Tag": []
        },
        "ImageVersion": "",
        "Status": "Available",
        "ImageName": "freebsd_11_02_64_30G_alibase_20190722.vhd",
        "IsSupportIoOptimized": true,
        "IsSelfShared": "",
        "IsCopied": false,
        "IsSubscribed": false,
        "Platform": "Freebsd",
        "Size": 30
      }
    ]
```

```
    }
}
```

- Obtain the query results of specific returned parameters, such as ImageId and ImageName:

```
freebsd_11_02_64_30G_alibase_20190722.vhd
freebsd_11_02_64_30G_alibase_20190722.vhd
```

Sample code

The following example shows complete Java SDK code.

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.alibaba.fastjson.JSON;
import java.util.*;
import com.aliyuncs.ecs.model.v20140526.*;

public class DescribeImages {

    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "LTAjVUwKznS*****", "BNPO1zoNS
i484oizGM9fzzwJJ*****");
        IAcsClient client = new DefaultAcsClient(profile);

        DescribeImagesRequest request = new DescribeImagesRequest();
        request.setRegionId("cn-hangzhou");
        request.setImageOwnerAlias("system");
        request.setPageNumber(1);
        request.setPageSize(1);

        try {
            DescribeImagesResponse response = client.getAcsResponse(request);
            System.out.println(JSON.toJSONString(response));
            for(DescribeImagesResponse.Image image:response.getImages())
            {
                System.out.println(image.getImageId());
                System.out.println(image.getImageName());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }

    }
}
```

Related information

- DescribeImages

2.3. Query available instance resources

This topic describes how to use the Alibaba Cloud ECS Java SDK to call the DescribeAvailableResource operation to query the list of available instance types in a zone. You can query available resources before you create ECS instances.

Prerequisites

You must use aliyun-java-sdk-ecs V4.6.3 or later.

Context

Before you create an ECS instance, you can call the DescribeAvailableResource operation to view the available instance resources in the specified region or zone. For more information, see DescribeAvailableResource. You can set object parameters such as region, zone, billing method, and instance family to query available instance resources. You can also set parameters of I/O optimization, system disk, and data disk to filter the query results.

Query the instance types available for subscription instances in a specific region

The following example demonstrates how to query available instance types to create subscription instances in the China (Hangzhou) region.

```
/**
 * Assume that you query the instance types that are available for subscription instances in the China (Hangzhou) region.
 * Target region: cn-hangzhou
 * If all zones are to be selected, zoneId is not specified.
 * Subscription: Set the InstanceChargeType parameter to PrePaid. Leave the SpotStrategy parameter unspecified or set the parameter to NoSpot.
 *
 * Note: If you set the DestinationResource parameter to InstanceType, you must specify the IoOptimized parameter.
 * Steps:
 * 1. Query available I/O optimized resources
 * 2. Query available instance types
 * Response:
 *   A list of instance types by zone ID is returned.
 *   If no instance types are available, null or an empty map is returned.
 */
public Map<String,Set<String>> doDescribeScene1() {
    DescribeAvailableResourceRequest describe = new DescribeAvailableResourceRequest();
    describe.setRegionId("cn-hangzhou");
```

```
        describe.setInstanceChargeType("PrePaid");
        // 1. Query available I/O optimized resources
        describe.setDestinationResource(IoOptimized);
        Map<String, Set<String>> ioOptimizeds = doActionAndProcessResponse(describe);
        if (null == ioOptimizeds) {
            return null;
        }
        // Set allTypes to specify a list of instance types, set the zone ID as the key, and enter the availa
ble instance types in the corresponding zone as the value.
        Map<String,Set<String>> allTypes = new HashMap<String, Set<String>>();
        for (String zoneId : ioOptimizeds.keySet()) {
            describe.setZoneId(zoneId);
            describe.setDestinationResource(InstanceType);
            for(String iopts : ioOptimizeds.get(zoneId)){
                describe.setIoOptimized(iopts);
            }

            Set<String> allTypesInZoneId = allTypes.get(zoneId);
            Map<String, Set<String>> types = doActionAndProcessResponse(describe);
            Set<String> typesInZoneId = types.get(zoneId);
            if(null ! = allTypesInZoneId){
                allTypesInZoneId.addAll(typesInZoneId);
            }else{
                allTypes.put(zoneId, typesInZoneId);
            }
        }
        return allTypes;
    }
```

Query the instance types available for pay-as-you-go instances in a specific region

The following example demonstrates how to query available instance types to create pay-as-you-go instances in the China (Hangzhou) region.

```
    /**
     Assume that you are querying the instance types that are available for pay-as-you-go instances in
the China (Hangzhou) region.
     * Target region: cn-hangzhou
     * If all zones are to be selected, zoneId is not specified.
     * Pay-as-you-go: Leave the InstanceChargeType parameter unspecified or set the parameter to Po
stPaid. Leave the SpotStrategy parameter unspecified or set the parameter to NoSpot.
```

```
    *
    * Note: If you set the DestinationResource parameter to InstanceType, you must specify the IoOpti
mized parameter.
    * Steps:
    * 1. Query available I/O optimized resources
    * 2. Query available instance types
    * Response:
    *    A list of instance types by zone ID is returned.
    *    If no instance types are available, null or an empty map is returned.
    */
   public Map<String,Set<String>> doDescribeScene2() {
      DescribeAvailableResourceRequest describe = new DescribeAvailableResourceRequest();
      describe.setRegionId("cn-hangzhou");
      describe.setInstanceChargeType("PostPaid");
      // 1. Query available I/O optimized resources
      describe.setDestinationResource(IoOptimized);
      Map<String, Set<String>> ioOptimizeds = doActionAndProcessResponse(describe);
      if (null == ioOptimizeds) {
         return null;
      }
      // Set allTypes to specify a list of instance types, set the zone ID as the key, and enter the availa
ble instance types in the corresponding zone as the value.
      Map<String,Set<String>> allTypes = new HashMap<String, Set<String>>();
      for (String zoneId : ioOptimizeds.keySet()) {
         describe.setZoneId(zoneId);
         describe.setDestinationResource(InstanceType);
         for(String iopts : ioOptimizeds.get(zoneId)){
            describe.setIoOptimized(iopts);
         }

         Set<String> allTypesInZoneId = allTypes.get(zoneId);
         Map<String, Set<String>> types = doActionAndProcessResponse(describe);
         Set<String> typesInZoneId = types.get(zoneId);
         if(null ! = allTypesInZoneId){
            allTypesInZoneId.addAll(typesInZoneId);
         }else{
            allTypes.put(zoneId, typesInZoneId);
         }
      }

      return allTypes;
```

```
    }
```

Query the instance types available for preemptible instances in a specific region

The following example demonstrates how to query available instance types to create preemptible instances in the China (Hangzhou) region.

```
    /**
     Assume that you are querying the instance types that are available for preemptible instances in th
e China (Hangzhou) region.
     * Target region: cn-hangzhou
     * If all zones are to be selected, zoneId is not specified.
     * Pay-as-you-go: Leave the InstanceChargeType parameter unspecified or set the parameter to Po
stPaid. Set the SpotStrategy parameter to SpotWithPriceLimit or SpotAsPriceGo.
     * Note: If you set the DestinationResource parameter to InstanceType, you must specify the IoOpti
mized parameter.
     * Steps:
     * 1. Query available I/O optimized resources
     * 2. Query available instance types
     * Response:
     *    A list of instance types by zone ID is returned.
     *    If no instance types are available, null or an empty map is returned.
     */
    public Map<String,Set<String>> doDescribeScene3() {
        DescribeAvailableResourceRequest describe = new DescribeAvailableResourceRequest();
        describe.setRegionId("cn-hangzhou");
        describe.setInstanceChargeType("PostPaid");
        describe.setSpotStrategy("SpotWithPriceLimit");
        // describe.setSpotStrategy("SpotAsPriceGo");
        // 1. Query available I/O optimized resources
        describe.setDestinationResource(IoOptimized);
        Map<String, Set<String>> ioOptimizeds = doActionAndProcessResponse(describe);
        if (null == ioOptimizeds) {
            return null;
        }
        // Set allTypes to specify a list of instance types, set the zone ID as the key, and enter the availa
ble instance types in the corresponding zone as the value.
        Map<String,Set<String>> allTypes = new HashMap<String, Set<String>>();
        for (String zoneId : ioOptimizeds.keySet()) {
            describe.setZoneId(zoneId);
            describe.setDestinationResource(InstanceType);
```

```
            describe.setDestinationResource(instanceType);
            for(String iopts : ioOptimizeds.get(zoneId)){
                describe.setIoOptimized(iopts);
            }

            Set<String> allTypesInZoneId = allTypes.get(zoneId);
            Map<String, Set<String>> types = doActionAndProcessResponse(describe);
            Set<String> typesInZoneId = types.get(zoneId);
            if(null ! = allTypesInZoneId){
                allTypesInZoneId.addAll(typesInZoneId);
            }else{
                allTypes.put(zoneId, typesInZoneId);
            }
        }

        return allTypes;
    }
```

Query the zones where a specific instance type is available

The following example demonstrates how to query the zones where a specific subscription instance type is available.

```
    /**
     * Assume that you are querying the zones where the ecs.gn4.8xlarge instance type is available for subscription instances.
     * Target region: cn-hangzhou
     * If all zones are to be selected, zoneId is not specified.
     * Target instance type: ecs.gn4.8xlarge
     * Subscription: Set the InstanceChargeType parameter to PrePaid. Leave the SpotStrategy parameter unspecified or set the parameter to NoSpot.
     *
     * Note: If you set the DestinationResource parameter to InstanceType, you must specify the IoOptimized parameter.
     * Steps:
     * 1. Query available I/O optimized resources
     * 2. Query available instance types
     * Response:
     *    The list of zones where the specified instance type is available
     *    If the specified instance type is unavailable in all zones, null or an empty value is returned.
     */
    public List<String> doDescribeScene4() {
```

```
        DescribeAvailableResourceRequest describe = new DescribeAvailableResourceRequest();
        describe.setRegionId("cn-hangzhou");
        describe.setInstanceChargeType("PrePaid");
        describe.setInstanceType("ecs.gn4.8xlarge");
        // 1. Query available I/O optimized resources
        describe.setDestinationResource(IoOptimized);
        Map<String, Set<String>> ioOptimizeds = doActionAndProcessResponse(describe);
        if (null == ioOptimizeds) {
            return null;
        }
        // Zones where the specified instance type is available
        List<String> zones = new ArrayList<String>(ioOptimizeds.size());
        for (String zoneId : ioOptimizeds.keySet()) {
            describe.setZoneId(zoneId);
            describe.setDestinationResource(InstanceType);
            for(String iopts : ioOptimizeds.get(zoneId)){
                describe.setIoOptimized(iopts);
            }
            Map<String, Set<String>> typesMap = doActionAndProcessResponse(describe);
            Set<String> types = typesMap.get(zoneId);
            if(CollectionUtils.isNotEmpty(types)){
                if(types.contains("ecs.gn4.8xlarge")){
                    zones.add(zoneId);
                }
            }
        }
        if(CollectionUtils.isNotEmpty(zones)){
            return zones;
        }
        return null;
    }
```

Query the instance types available for VPC-type subscription instances in a specific zone

The following example demonstrates how to query available instance types to create VPC-type subscription instances in the China (Hangzhou) region.

```
    /**
     * Assume that you are querying the instance types that are available for VPC-type subscription ins
tances in Hangzhou Zone E.
     * Target region: cn-hangzhou
```

```
    * Target zone: cn-hangzhou-e
    * Target network type: VPC
    * Subscription: Set the InstanceChargeType parameter to PrePaid. Leave the SpotStrategy parame
ter unspecified or set the parameter to NoSpot.
    *
    * Note: If you set the DestinationResource parameter to InstanceType, you must specify the IoOpti
mized parameter.
    * Steps:
    * 1. Query available I/O optimized resources
    * 2. Query available instance types
    * Response:
    *    The list of qualified instance types that are available in the specified zone.
    *    If no qualified instance type is available in the specified zone, null or an empty value is returned
.
    */
   public List<String> doDescribeScene5() {
      DescribeAvailableResourceRequest describe = new DescribeAvailableResourceRequest();
      describe.setRegionId("cn-hangzhou");
      describe.setZoneId("cn-hangzhou-e");
      describe.setInstanceChargeType("PrePaid");
      describe.setNetworkCategory("Vpc");
      // Query available I/O optimized resources
      describe.setDestinationResource(IoOptimized);
      Map<String, Set<String>> ioOptimizedMap = doActionAndProcessResponse(describe);
      if (null == ioOptimizedMap) {
         return null;
      }
      Set<String> ioOptimizeds = ioOptimizedMap.get("cn-hangzhou-e");
      if(CollectionUtils.isEmpty(ioOptimizeds)) {
         return null;
      }
      // The instance types that meet the specifications
      Set<String> types = new HashSet<String>();
      describe.setDestinationResource(InstanceType);

      for(String iopts : ioOptimizeds){
         describe.setIoOptimized(iopts);
         Map<String, Set<String>> typesMap = doActionAndProcessResponse(describe);
         Set<String> typesInMap = typesMap.get("cn-hangzhou-e");
         if(CollectionUtils.isNotEmpty(typesInMap)){
            types.addAll(typesInMap);
```

```
        }
    }
    if(CollectionUtils.isNotEmpty(types)){
        return new ArrayList<String>(types);
    }
    return null;
}
```

What's next

Batch create ECS instances

Related information

- DescribeAvailableResource

2.4. Create an ECS instance

2.4.1. Batch create ECS instances

This topic describes how to use the Alibaba Cloud ECS Java SDK to call the RunInstances operation to create one or more ECS instances.

Prerequisites

You must query the following information before you create ECS instances:

- Call the DescribeRegions operation to query the region where you want to create ECS instances. In this example, the region is *cn-hangzhou*.
- Call the DescribeImages operation to query the ID of the image that you want to use. In this example, the ID of the image is *freebsd_11_02_64_30G_alibase_20190722.vhd*.
- Call the DescribeInstanceTypes operation to query the instance type that you want to select. In this example, the instance type is *ecs.g5.large*. For more information, see Instance families.
- Call the DescribeSecurityGroups operation to query IDs of one or more security groups in the specified region. In this example, the ID of the security group is *sg-bp1fg655nh68xyz9i****. The network type of the security group determines the network type of the ECS instance. For example, if you choose a security group in a VPC, the new ECS instance is automatically added to the VPC.
- If the security group is in a VPC, call the VPC API DescribeVSwitches operation to query the ID of the VSwitch in the VPC. In this example, the ID of the VSwitch is *vsw-bp1wt4qpuavdb6y6k8****.

Context

The following section demonstrates how to call the RunInstances operation to batch create and start ECS instances. For more information, see RunInstances.

Note If you call the RunInstances operation, billable resources such as ECS instances can be created and incur fees. If you need to test the sample code, you can set the DryRun parameter in the code to send check requests without creating instances. Check items include whether the required parameters are set, and verifies the request format, service limits, and available ECS instances.

Sample code

The following code can be used to create pay-as-you-go ECS instances that use the pay-by-traffic billing method for network usage in a VPC:

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.alibaba.fastjson.JSON;
import java.util.*;
import java.util.UUID;
import com.aliyuncs.ecs.model.v20140526.*;

public class RunInstances {

   public static void main(String[] args) {
      // Create and initialize a DefaultAcsClient instance.
      DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<yourAccessKeyId>", "<yourAcce
ssSecret>");
      IAcsClient client = new DefaultAcsClient(profile);

      // Create an API request and configure the parameters.
      RunInstancesRequest request = new RunInstancesRequest();
      request.setRegionId("cn-hangzhou");
      request.setImageId("freebsd_11_02_64_30G_alibase_20190722.vhd");
      request.setInstanceType("ecs.g5.large");
      request.setSecurityGroupId("sg-bp1fg655nh68xyz9i***");
      request.setVSwitchId("vsw-bp1wt4qpuavdb6y6k8***");
      request.setInstanceName("MyFirstEcsInstance");
      request.setDescription("MyFirstEcsInstance");
      request.setInternetMaxBandwidthOut(2);
      request.setInternetChargeType("PayByTraffic");
      request.setClientToken(UUID.randomUUID().toString());

      // Add a 100 GiB standard SSD data disk, and enable the Release Disk with Instance feature for th
```

```
        // Add a 100 GiB standard SSD data disk, and enable the Release Disk with Instance feature for th
e disk.
        List<RunInstancesRequest.DataDisk> dataDiskList = new ArrayList<RunInstancesRequest.DataDis
k>();

        RunInstancesRequest.DataDisk dataDisk1 = new RunInstancesRequest.DataDisk();
        dataDisk1.setSize(100);
        dataDisk1.setCategory("cloud_ssd");
        dataDisk1.setDeleteWithInstance(true);
        dataDiskList.add(dataDisk1);
        request.setDataDisks(dataDiskList);
        // Batch create five ECS instances. If the Amount parameter is not configured, one ECS instance wi
ll be created.
        // request.setAmount(5);
        // The minimum number of instance to create if available resources are insufficient.
        // request.setMinAmount(2);

        List<RunInstancesRequest.Tag> tagList = new ArrayList<RunInstancesRequest.Tag>();

        RunInstancesRequest.Tag tag1 = new RunInstancesRequest.Tag();
        tag1.setKey("EcsProduct");
        tag1.setValue("DocumentationDemo");
        tagList.add(tag1);
        request.setTags(tagList);
        // If you enable the precheck parameter function, the system will not create an ECS instance and
only check parameter correctness, user permissions, or ECS resource inventory.
        // If the DryRun parameter is configured to true, Amount must be 1 and MinAmount must be empty
. You can modify the code as needed.
        // request.setDryRun(true);
        request.setInstanceChargeType("PostPaid");

        // Initiate the request and handle the response or exceptions.
        RunInstancesResponse response;
        try {
            response = client.getAcsResponse(request);

            System.out.println(JSON.toJSONString(response));

        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
```

```
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }

    }
}
```

Result

Response:

```
{
    "RequestId":"04F0F334-1335-436C-A1D7-6C044FE73368",
    "InstanceIdSets":{
        "InstanceIdSet":[
            "i-instanceid1",
            "i-instanceid2",
            "i-instanceid3"
        ]
    }
}
```

Related information

- RunInstances
- DescribeRegions
- DescribeImages
- DescribeInstanceTypes
- DescribeSecurityGroups
- DescribeVSwitches

2.4.2. Create a preemptible instance

This topic describes how to use the Alibaba Cloud ECS Java SDK to call the DescribeAvailableResource operation to query available resources, call the DescribeSpotPriceHistory operation to query the historical prices of preemptible instances, and call the CreateInstance operation to create a preemptible instance.

Prerequisites

- The Alibaba Cloud ECS Java SDK that you use must be V4.2.0 or later.
- The image used to create a preemptible instance must contain all of the environment elements required to run the instance. We recommend that you use a custom image to create a preemptible instance to ensure that the instance can process business data. In this example, the ID of the image is *m-bp146shijn7hujkui9****.
- Call the DescribeRegions operation to query the region where you want to create ECS

instances. In this example, the region is *cn-hangzhou*.

- Call the DescribeInstanceTypes operation to query the instance type that you want to select. In this example, the instance type is *ecs.g5.large*. For more information, see Instance families.

Context

Preemptible instances provide a cost-effective solution to process business data. When you create a preemptible instance, you can set a maximum price per hour to bid for a specified instance type. If your price is higher than the current market price, your instance will be created and billed at the current market price. Preemptible instances are often used to control computing costs and are suitable in stateless application scenarios. For more information, see Overview.

Procedure

1. Encapsulate an ApiCaller.java class to initialize the profile and client and add exception handling logic.

```
public class ApiCaller {
    IClientProfile profile;
    IAcsClient client;

    public ApiCaller() {
        profile = DefaultProfile.getProfile("cn-hangzhou", AKSUtil.accessKeyId, AKSUtil.accessKeySec
ret);
        client = new DefaultAcsClient(profile);
    }

    public  <T extends AcsResponse> T doAction(AcsRequest<T> var1) {
        try {
            return  client.getAcsResponse(var1);
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }

}
```

2. Call the DescribeAvailableResource operation to query the instance types available for the specific region.

```
public class DescribeAvailableResourceSample {

    public static void main(String[] args) {

        ApiCaller caller = new ApiCaller();
        DescribeAvailableResourceRequest request = new DescribeAvailableResourceRequest();
        // Call the DescribeRegionsRequest operation to obtain the region ID.
        request.setRegionId("cn-hangzhou");
        request.setDestinationResource("InstanceType");
        // Set the billing method to pay-as-you-go. This parameter is required.
        request.setInstanceChargeType("PostPaid");
        // Set the bidding policy. This parameter is required.
        request.setSpotStrategy("SpotAsPriceGo");
        // If no zone ID is specified, all zones in which preemptible instances can be created will be q
ueried.
        request.setZoneId("cn-hangzhou-h");
        // If no instance type is specified, all instance types that can be selected to create preemptib
le instances will be queried.
        request.setInstanceType("ecs.g5.large");
        request.setSystemDiskCategory("cloud_ssd");
        request.setNetworkCategory("vpc");

        DescribeAvailableResourceResponse response = caller.doAction(request);
        System.out.println(JSON.toJSONString(response));
    }
}
```

Sample response:

```
{
  "RequestId": "D8491D5E-AB8A-4E22-BDB4-EEEE1F1C8241",
  "AvailableZones": {
    "AvailableZone": [
      {
        "Status": "Available",
        "RegionId": "cn-hangzhou",
        "AvailableResources": {
          "AvailableResource": [
            {
              "Type": "InstanceType",
              "SupportedResources": {
                "SupportedResource": [
                  {
                    "Status": "Available",
                    "Value": "ecs.g5.large",
                    "StatusCategory": "WithStock"
                  }
                ]
              }
            }
          ]
        },
        "ZoneId": "cn-hangzhou-h",
        "StatusCategory": "WithStock"
      }
    ]
  }
}
```


3. (Optional) Call the DescribeSpotPriceHistory operation to query the historical prices of preemptible instances.

> **Note** DescribeSpotPriceHistory allows you to obtain the historical prices for up to the last 30 days.

```
public class DescribeSpotPriceHistorySample {

  public static void main(String[] args) {

    ApiCaller caller = new ApiCaller();
    List<DescribeSpotPriceHistoryResponse.SpotPriceType> result = new ArrayList<DescribeSpot
```

```
        List<DescribeSpotPriceHistoryResponse.SpotPriceType> result = new ArrayList<DescribeSpot
PriceHistoryResponse.SpotPriceType>();
        int offset = 0;
        while (true) {
            DescribeSpotPriceHistoryRequest request = new DescribeSpotPriceHistoryRequest();
            request.setRegionId("cn-hangzhou");// The region is required.
            request.setZoneId("cn-hangzhou-b");// The zone is required.
            request.setInstanceType("ecs.g5.large");// The instance type is required.
            request.setNetworkType("vpc");// The network type is required.
            // request.setStartTime("2017-09-20T08:45:08Z");// Optional. The start time of a time perio
d for which the historical prices of preemptible instances are to be queried. The time period is thr
ee days by default.
            // request.setEndTime("2017-09-28T08:45:08Z");// Optional. The end time of a time period f
or which the historical prices of preemptible instances are to be queried.
            request.setOffset(offset);
            DescribeSpotPriceHistoryResponse response = caller.doAction(request);
            if (response !\ = null && response.getSpotPrices() ! = null) {
                result.addAll(response.getSpotPrices());
            }
            if (response.getNextOffset() == null || response.getNextOffset() == 0) {
                break;
            } else {
                offset = response.getNextOffset();
            }
        }
        if (! result.isEmpty()) {
            for (DescribeSpotPriceHistoryResponse.SpotPriceType spotPriceType : result) {
                System.out.println(spotPriceType.getTimestamp() + "--->spotPrice:" + spotPriceType.get
SpotPrice() + "---->originPrice:" + spotPriceType.getOriginPrice());
            }
            System.out.println(result.size());
        } else {
        }

    }

}
```

Sample response:

```
2017-09-26T06:28:55Z--->spotPrice:0.24---->originPrice:1.2
2017-09-26T14:00:00Z--->spotPrice:0.36---->originPrice:1.2
2017-09-26T15:00:00Z--->spotPrice:0.24---->originPrice:1.2
2017-09-27T14:00:00Z--->spotPrice:0.36---->originPrice:1.2
2017-09-27T15:00:00Z--->spotPrice:0.24---->originPrice:1.2
2017-09-28T14:00:00Z--->spotPrice:0.36---->originPrice:1.2
2017-09-28T15:00:00Z--->spotPrice:0.24---->originPrice:1.2
2017-09-29T06:28:55Z--->spotPrice:0.24---->originPrice:1.2
```

4. Call the CreateInstances operation to create a preemptible instance.

   You can call the RunInstance operation to create multiple preemptible instances at a time. For more information, see Batch create ECS instances.

```
public class CreateInstancesSample {

    public static void main(String[] args) {
        ApiCaller caller = new ApiCaller();
        CreateInstancesRequest request = new CreateInstancesRequest();
        request.setRegionId("cn-hangzhou");// The ID of the region.
        request.setZoneId("cn-hangzhou-h"); // The ID of the zone.
        request.setSecurityGroupId("sg-bp11nhf94ivkdxwb2***");// The ID of the security group.
        request.setImageId("m-bp146shijn7hujkui9***");// We recommend that you use a custom imag
e to create instances more quickly.
        request.setVSwitchId("vsw-bp164cyonthfudn9kj***");// The ID of the VSwitch that is in the sa
me VPC as the security group.

        request.setInstanceType("ecs.g5.large"); // Enter the instance type you want to purchase aft
er the query.
        request.setSystemDiskCategory("cloud_ssd");// The category of the system disk. Valid value
s: cloud_essd, cloud_ssd, cloud_efficiency, and cloud.
        request.setSystemDiskSize(40);

        request.setInstanceChargeType("PostPaid");// Set the billing method to pay-as-you-go. This
parameter is required.
        request.setSpotStrategy("SpotWithPriceLimit");// SpotWithPriceLimit: the bidding mode. Man
ually set the maximum price. SpotAsPriceGo: The system provides a price automatically based on t
he market price.
        request.setSpotPriceLimit(0.25F);// If the bidding mode is set to SpotWithPriceLimit, the maxi
mum price must be set higher than the current market price to create the instance.
        // request.setAmount(2);// Create two preemptible instances. If the parameter is not configu
red, only one preemptible instance will be created.

        CreateInstanceResponse response = caller.doAction(request);
        System.out.println(response.getInstanceId());

    }
}
```

5. Recycle a preemptible instance

> **Note** After the instance is created, you must make sure to complete all of your business within the guaranteed duration and prepare for inconvenience that may be caused by the recycle of the instance.

i. Run the following command in the operating system of the preemptible instance to query the recycle time of the instance based on the instance metadata:

```
curl 'http://100.100.100.200/latest/meta-data/instance/spot/termination-time'
```

> **Note** If the response is empty, the preemptible instance can continue to be used. If information in format of `2015-01-05T18:02:00Z` (UTC+0) is returned in the response, the preemptible instance will be recycled at that point of time.

ii. Call the DescribeInstances operation to determine whether the instance has entered the Pending Release state based on the returned OperationLocks parameter.

```
public class DescribeInstancesSample {
  public static void main(String[] args) throws InterruptedException {
    ApiCaller caller = new ApiCaller();
    JSONArray allInstances = new JSONArray();
    allInstances.addAll(Arrays.asList("i-bp18hgfai8ekoqwo0***", "i-bp1ecbyds24ij63w1***"));
    while (! allInstances.isEmpty()) {
      DescribeInstancesRequest request = new DescribeInstancesRequest();
      request.setRegionId("cn-hangzhou");
      request.setInstanceIds(allInstances.toJSONString());// Specify the ID of the instance an
d only query the instance whose ID has been specified.
      DescribeInstancesResponse response = caller.doAction(request);
      List<DescribeInstancesResponse.Instance> instanceList = response.getInstances();
      if (instanceList ! = null && ! instanceList.isEmpty()) {
        for (DescribeInstancesResponse.Instance instance : instanceList) {
          System.out.println("result:instance:" + instance.getInstanceId() + ",was created in
zone:" + instance.getZoneId());
          if (instance.getOperationLocks() ! = null) {
            for (DescribeInstancesResponse.Instance.LockReason lockReason : instance.get
OperationLocks()) {
              System.out.println("instance: " + instance.getInstanceId() + "-->lockReason:" +
lockReason.getLockReason() + ",vmStatus:" + instance.getStatus());
              if ("Recycling".equals(lockReason.getLockReason())) {
                // Return the recycle time of the instance and delete the ID of the recycled in
stance from the instance ID list.
                System.out.println("Preemptible instance will be recycled immediately, insta
nce id: " + instance.getInstanceId());
                allInstances.remove(instance.getInstanceId());
              }
            }
          }
        }
        System.out.print("Try describeInstancesRequest again later...") ;
        Thread.sleep(2 * 60 * 1000);
      } else {
        break;
      }
    }
  }
}
```

If the following response is returned, the preemptible instance has entered the Pending Release state:

```
instance: i-bp1ecbyds24ij63w1***-->lockReason:Recycling,vmStatus:Stopped
Preemptible instance will be recycled immediately, instance id: i-bp1ecbyds24ij63w1***
```

Related information

- DescribeRegions
- DescribeInstances
- DescribeAvailableResource
- DescribeInstanceTypes
- DescribeSpotPriceHistory
- CreateInstance
- RunInstances

2.5. Query an ECS instance

This topic describes how to use the Alibaba Cloud ECS Java SDK to call the DescribeInstances operation to filter ECS instances based on specified conditions.

Prerequisites

You must have at least one existing ECS instance. For more information, see Batch create ECS instances.

Context

You can call this operation to query instances that meet specified conditions. For example:

- Before you modify the public bandwidth of an instance with a specific ID, you can filter eligible ECS instances based on the billing method of instances, running status, and billing method for network usage.
- Before you update applications on an instance, query all ECS instances that use the same image as the instance.

Sample code

The following code can be used to query VPC-type pay-as-you-go instances that use pay-by-traffic billing method for network usage in the China (Hangzhou) region.

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import java.util.*;
import com.aliyuncs.ecs.model.v20140526.*;

public class DescribeInstances {
```

```
    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<yourAccessKeyId>", "<yourAcce
ssSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        // Create an API request and configure the parameters.
        DescribeInstancesRequest request = new DescribeInstancesRequest();
        request.setRegionId("cn-hangzhou");
        request.setInstanceNetworkType("vpc");
        request.setInstanceChargeType("PostPaid");
        request.setInternetChargeType("PayByTraffic");
        request.setPageSize(10);

        try {
        // Initiate the request and handle the response or exceptions.
            DescribeInstancesResponse response = client.getAcsResponse(request);
        for (DescribeInstancesResponse.Instance instance:response.getInstances())
            {
                System.out.println(instance.getImageId());
                System.out.println(instance.getInstanceId());
                System.out.println(instance.getPublicIpAddress());
             }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }

    }
}
```

Result

The preceding code requests the ID of the image, the ID of the instance, and the public IP address. The response is:

```
i-bp1gvi17n5p8hav0i***
[47.97. ***.21]
ubuntu_16_04_64_20G_alibase_20190620.vhd
i-bp1gc5z6103qs2t40***
[47.99. ***.82]
centos_7_06_64_20G_alibase_20190711.vhd
```

Related information

- DescribeInstances

2.6. Change the instance type of an ECS instance

This topic describes how to use the Alibaba Cloud ECS Java SDK to call the ModifyInstanceSpec operation to change the instance type of a pay-as-you-go instance and call the ModifyPrepayInstanceSpec operation to change the instance type of a subscription instance.

Prerequisites

When you change the instance type of an ECS instance, note that:

- You must not have overdue payments for the pay-as-you-go instance.
- The subscription instance cannot be in the Expired state.
- The instance must be in the Stopped state.

Change the instance type of a subscription instance

The following code can be used to call the ModifyPrepayInstanceSpec operation to change the instance type of a subscription instance to ecs.g5.large in the China (Hangzhou) region:

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import java.util.*;
import com.aliyuncs.ecs.model.v20140526.*;

public class ModifyPrepayInstanceSpec {

    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<yourAccessKeyId>", "<yourAcce
ssSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and configure the parameters.
        ModifyPrepayInstanceSpecRequest request = new ModifyPrepayInstanceSpecRequest();
        request.setRegionId("cn-hangzhou");
        request.setInstanceId("i-bp1jd3uddaduyo8*****");
        // Set the new instance type. ModifyPrepayInstanceSpecRequest allows you to upgrade and dow
ngrade the instance type.
        request.setInstanceType("ecs.g5.large");

        try {
            ModifyPrepayInstanceSpecResponse response = client.getAcsResponse(request);
            logInfo(response.getOrderId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }

    }
    private static void logInfo(String message) {
        System.out.println(message);
    }
}
```

Change the instance type of a pay-as-you-go instance

The following code can be used to call the ModifyInstanceSpec operation to change the instance type of a pay-as-you-go instance to ecs.g5.large in the China (Hangzhou) region:

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.alibaba.fastjson.JSON;
import java.util.*;
import com.aliyuncs.ecs.model.v20140526.*;

public class ModifyInstanceSpec {

    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<yourAccessKeyId>", "<yourAcce
ssSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and configure the parameters.
        ModifyInstanceSpecRequest request = new ModifyInstanceSpecRequest();
        request.setRegionId("default");
        // You must specify the instance ID.
        request.setInstanceId("i-bp1gc5z6103qs2t40***");
        // Set the new instance type. ModifyInstanceSpecRequest allows you to upgrade and downgrade
the instance type.
        request.setInstanceType("ecs.g5.large");

        try {
            ModifyInstanceSpecResponse response = client.getAcsResponse(request);
            System.out.println(JSON.toJSONString(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }

    }
}
```

Related information

- ModifyInstanceSpec
- ModifyPrepayInstanceSpec

2.7. Release an ECS instance

This topic describes how to use the Alibaba Cloud ECS Java SDK to call the DeleteInstance operation to delete an ECS instance.

Alibaba Cloud ecs java sdk release an instance.

Prerequisites

You must have at least one existing ECS instance. For more information, see Batch create ECS instances.

Sample code

The following code can be used to delete an ECS instance in China (Hangzhou).

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.ecs.model.v20140526.*;

public class DeleteInstance {

   public static void main(String[] args) {
      // Create and initialize a DefaultAcsClient instance.
      DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<yourAccessKeyId>", "<yourAcce
ssSecret>");
      IAcsClient client = new DefaultAcsClient(profile);

      // Create an API request and configure the parameters.
      DeleteInstanceRequest request = new DeleteInstanceRequest();
      // Specify a region ID.
      request.setRegionId("cn-hangzhou");
      // Specify an instance ID.
      request.setInstanceId("i-bp1gvi17n5p8hav0i***");

      try {
      // Initiate the request and handle the response or exceptions.
         DeleteInstanceResponse response = client.getAcsResponse(request);
         System.out.println(new Gson().toJson(response));
      } catch (ServerException e) {
         e.printStackTrace();
      } catch (ClientException e) {
         System.out.println("ErrCode:" + e.getErrCode());
         System.out.println("ErrMsg:" + e.getErrMsg());
         System.out.println("RequestId:" + e.getRequestId());
      }

   }
}
```

Result

Response:

```
{"RequestId":"928E2273-5715-46B9-A730-238DC996A533"}
```

Related information

- DeleteInstance
- DeleteInstances

3.Python example

3.1. Create multiple ECS instances at a time

You can call RunInstances to create multiple pay-as-you-go or subscription ECS instances. This allows you to scale resources with ease to meet requirements of developing and deploying applications.

Prerequisites

An AccessKey pair is created before you call the API operation. For more information about the procedure, see .

> **Notice** Do not use the AccessKey pair of your Alibaba Cloud account. If the AccessKey pair is leaked, the security of all your resources is threatened. We recommend that you use the AccessKey pair of a RAM user to reduce the risk of leakage.

Context

Compared with CreateInstance, RunInstances has the following benefits:

- A maximum of 100 ECS instances can be created at a time.
- After an ECS instance is created, the status of the instance automatically becomes *Starting* and then *Running*. You do not need to call the StartInstance operation.
- If you specify InternetMaxBandwidthOut when you create an ECS instance, a public IP address is automatically assigned to the instance. You do not need to call an API operation to assign a public IP address to the instance.
- You can create up to 100 preemptible instances at a time to meet your business requirements.
- The parameters of RunInstances are compatible with those of CreateInstance. The Amount parameter is added to specify the number of ECS instances that you want to create. The AutoReleaseTime parameter is added to specify the automatic release time.
- After ECS instances are created, InstanceIdSets that contains InstanceIds is returned. You can query the status of the instances based on their IDs. For more information, see DescribeInstanceStatus.

This topic provides the sample code to show how to create multiple ECS instances at a time. For more information, see Sample code. This topic also provides the explanation of the sample code. For more information, see the following topics:

- Create multiple ECS instances at a time
- Create multiple ECS instances and assign public IP addresses
- Create multiple ECS instances and set their automatic release time

Install ECS SDK for Python

Ensure that you have prepared the runtime environment for Python. Python 2.7 or later is used in this topic.

SDK for Python 4.4.3 is used in this topic. If you are using an earlier version, update it.

```
pip install aliyun-python-sdk-ecs
```

If you are prompted that you do not have the permission, use the following **sudo** command to continue the installation:

```
sudo pip install aliyun-python-sdk-ecs
```

Sample code

```
#  coding=utf-8
# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 4.4.3, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging
import time
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.RunInstancesRequest import RunInstancesRequest
logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',
                    datefmt='%a, %d %b %Y %H:%M:%S')

# Your AccessKey ID.
ak_id = "your-access-key-id"

# Your AccessKey secret.
ak_secret = "your-access-key-secret"

# The region ID of the ECS instances.
region_id = "cn-hangzhou"

clt = client.AcsClient(ak_id, ak_secret, region_id)

# The instance type.
instance_type = "ecs.g6.large"

# The ID of the selected VSwitch.
vswitch_id = "vsw-bp1ddbrxdlrcbim46****"
```

```
# The ID of the image that you use.
image_id = "<imageid>"

# The ID of the security group in the current VPC.
security_group_id = "sg-bp1i4c0xgqxadew2****"

# The number of ECS instances that you want to create. Valid values: 1 to 100. Default value: 1.
amount = 2;

# The automatic release time. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ss
Z format. The time must be in UTC. The release time ranges from 30 minutes after the current time to t
hree years later than the current time.
auto_release_time = "2020-04-17T18:20:00Z"

# Create and start ECS instances.
def create_multiple_instances():
    request = build_request()
    request.set_Amount(amount)
    _execute_request(request)

# Create ECS instances and assign public IP addresses.
def create_multiple_instances_with_public_ip():
    request = build_request()
    request.set_Amount(amount)
    request.set_InternetMaxBandwidthOut(1)
    _execute_request(request)

# Create ECS instances and set their automatic release time.
def create_multiple_instances_with_auto_release_time():
    request = build_request()
    request.set_Amount(amount)
    request.set_AutoReleaseTime(auto_release_time)
    _execute_request(request)

def _execute_request(request):
    response = _send_request(request)
    if response.get('Code') is None:
        instance_ids = response.get('InstanceIdSets').get('InstanceIdSet')
        running_amount = 0
        while running_amount < amount:
            time.sleep(10)
```

```
        running_amount = check_instance_running(instance_ids)
    print("ecs instance %s is running", instance_ids)

def check_instance_running(instance_ids):
    request = DescribeInstancesRequest()
    request.set_InstanceIds(json.dumps(instance_ids))
    response = _send_request(request)
    if response.get('Code') is None:
        instances_list = response.get('Instances').get('Instance')
        running_count = 0
        for instance_detail in instances_list:
            if instance_detail.get('Status') == "Running":
                running_count += 1
        return running_count

def build_request():
    request = RunInstancesRequest()
    request.set_ImageId(image_id)
    request.set_VSwitchId(vswitch_id)
    request.set_SecurityGroupId(security_group_id)
    request.set_InstanceName("Instance-Name")
    request.set_InstanceType(instance_type)
    return request

# Initiate an API request.
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)

if __name__ == '__main__':
    print ("hello ecs batch create instance")
    # Create and start ECS instances.
    create_multiple_instances()
    # Create ECS instances and bind public IP addresses.
```

```
# create_multiple_instances_with_public_ip()
# Create ECS instances and set their automatic release time.
# create_multiple_instances_with_auto_release_time()
```

Create multiple ECS instances at a time

You can specify the required parameters to create multiple ECS instances at a time.

In the following example, two ECS instances are created. The status of the instances is automatically checked every 10 seconds. The creation is completed after the ECS instances enter the *Running* state.

```
# Your AccessKey ID.
ak_id = "your-access-key-id"

# Your AccessKey secret.
ak_secret = "your-access-key-secret"

# The region ID of the ECS instances.
region_id = "cn-hangzhou"

clt = client.AcsClient(ak_id, ak_secret, region_id)

# The instance type.
instance_type = "ecs.g6.large"

# The ID of the selected VSwitch.
vswitch_id = "vsw-bp1ddbrxdlrcbim46****"

# The ID of the image that you use.
image_id = "<imageid>"

# The ID of the security group in the current VPC.
security_group_id = "sg-bp1i4c0xgqxadew2****"

# The number of ECS instances that you want to create. Valid values: 1 to 100. Default value: 1.
amount = 2;

# The automatic release time. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ss
Z format. The time must be in UTC. The earliest release time is 30 minutes after the current time. The l
atest release time cannot be three years later than the current time.
auto_release_time = "2020-04-17T18:20:00Z"
```

```
# Create and start ECS instances.
def create_multiple_instances():
    request = build_request()
    request.set_Amount(amount)
    _execute_request(request)

def _execute_request(request):
    response = _send_request(request)
    if response.get('Code') is None:
        instance_ids = response.get('InstanceIdSets').get('InstanceIdSet')
        running_amount = 0
        while running_amount < amount:
            time.sleep(10)
            running_amount = check_instance_running(instance_ids)
    print("ecs instance %s is running", instance_ids)

def check_instance_running(instance_ids):
    request = DescribeInstancesRequest()
    request.set_InstanceIds(json.dumps(instance_ids))
    response = _send_request(request)
    if response.get('Code') is None:
        instances_list = response.get('Instances').get('Instance')
        running_count = 0
        for instance_detail in instances_list:
            if instance_detail.get('Status') == "Running":
                running_count += 1
        return running_count

def build_request():
    request = RunInstancesRequest()
    request.set_ImageId(image_id)
    request.set_VSwitchId(vswitch_id)
    request.set_SecurityGroupId(security_group_id)
    request.set_InstanceName("Instance-Name")
    request.set_InstanceType(instance_type)
    return request

# Initiate an API request.
def _send_request(request):
    request.set_accept_format('json')
    try:
```

```
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)
```

Create multiple ECS instances and assign public IP addresses

You can specify the public bandwidth when you create multiple ECS instances. In the following example, the instances are assigned with a bandwidth of 1 Mbit/s:

```
# Create ECS instances and assign public IP addresses.
def create_multiple_instances_with_public_ip():
    request = build_request()
    request.set_Amount(amount)
    request.set_InternetMaxBandwidthOut(1)
    _execute_request(request)
```

Create multiple ECS instances and set their automatic release time

You can specify the automatic release time when you create multiple ECS instances. Specify the time in the ISO 8601 standard in the `yyyy-MM-ddTHH:mm:ssZ` format. The time must be in UTC. The release time ranges from 30 minutes after the current time to three years later than the current time. For more information, see ISO 8601 Time Format.

```
# Create ECS instances and set their automatic release time.
def create_multiple_instances_with_auto_release_time():
    request = build_request()
    request.set_Amount(amount)
    request.set_AutoReleaseTime(auto_release_time)
    _execute_request(request)
```

3.2. Create an ECS instance

This topic describes how to create and manage an ECS instance by using Alibaba Cloud API operations.

Context

This topic provides the sample code to show how to create an ECS instance and also the explanation of the sample code. For more information, see the following topics:

- Create a pay-as-you-go ECS instance
- Check and start an ECS instance after creation

- Assign a public IP address to an ECS instance
- Create a subscription ECS instance

Sample code

The following content is the sample code for creating an instance. You can modify the code and change the values of parameters based on your actual needs.

```
#  coding=utf-8
# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 4.4.3, you can use command 'pip show aliyun-python-sdk-ecs' to check
import json
import logging
import time
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.CreateInstanceRequest import CreateInstanceRequest
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.StartInstanceRequest import StartInstanceRequest
# configuration the log output formatter, if you want to save the output to file,
# append ",filename='ecs_invoke.log'" after datefmt.

logging.basicConfig(level=logging.INFO, format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelna
me)s %(message)s', datefmt='%a, %d %b %Y %H:%M:%S')

clt = client.AcsClient('<accessKeyId>', '<accessSecret>', '<region-Id>')
IMAGE_ID = 'ubuntu_18_04_64_20G_alibase_20190624.vhd'
INSTANCE_TYPE = 'ecs.g6.large'
SECURITY_GROUP_ID = 'sg-bp1i4c0xgqxadew2****'
INSTANCE_RUNNING = 'Running'
VSWITCH_ID = 'vsw-bp1ddbrxdlrcbim46****'

def create_instance_action():
    instance_id = create_after_pay_instance(IMAGE_ID, INSTANCE_TYPE, SECURITY_GROUP_ID, VSWITCH_I
D)
    check_instance_running(instance_id)

# def create_prepay_instance_action():
#     instance_id = create_prepay_instance(IMAGE_ID, INSTANCE_TYPE, SECURITY_GROUP_ID, VSWITCH_I
D)
#     check_instance_running(instance_id=instance_id)

# create one after pay ecs instance.
```

```
def create_after_pay_instance(image_id, instance_type, security_group_id, vsw_vswitch_id):
    request = CreateInstanceRequest()
    request.set_ImageId(image_id)
    request.set_SecurityGroupId(security_group_id)
    request.set_InstanceType(instance_type)
    request.set_IoOptimized('optimized')
    request.set_VSwitchId(vsw_vswitch_id)
    request.set_SystemDiskCategory('cloud_ssd')
    response = _send_request(request)
    instance_id = response.get('InstanceId')
    logging.info("instance %s created task submit successfully.", instance_id)
    return instance_id

# create one prepay ecs instance.
def create_prepay_instance(image_id, instance_type, security_group_id, vsw_vswitch_id):
    request = CreateInstanceRequest()
    request.set_ImageId(image_id)
    request.set_SecurityGroupId(security_group_id)
    request.set_InstanceType(instance_type)
    request.set_IoOptimized('optimized')
    request.set_VSwitchId(vsw_vswitch_id)
    request.set_SystemDiskCategory('cloud_ssd')
    request.set_Period(1)
    request.set_InstanceChargeType('PrePaid')
    response = _send_request(request)
    instance_id = response.get('InstanceId')
    logging.info("instance %s created task submit successfully.", instance_id)
    return instance_id

def check_instance_running(instance_id):
    detail = get_instance_detail_by_id(instance_id, INSTANCE_RUNNING)
    index = 0
    while detail is None and index < 60:
        detail = get_instance_detail_by_id(instance_id)
        time.sleep(10)
    if detail and detail.get('Status') == 'Stopped':
        logging.info("instance %s is stopped now.")
        start_instance(instance_id)
        logging.info("start instance %s job submit.")
    detail = get_instance_detail_by_id(instance_id, INSTANCE_RUNNING)
```

```
    while detail is None and index < 60:
        detail = get_instance_detail_by_id(instance_id, INSTANCE_RUNNING)
        time.sleep(10)
    logging.info("instance %s is running now.", instance_id)
    return instance_id

def start_instance(instance_id):
    request = StartInstanceRequest()
    request.set_InstanceId(instance_id)
    _send_request(request)

# output the instance owned in current region.
def get_instance_detail_by_id(instance_id, status='Stopped'):
    logging.info("Check instance %s status is %s", instance_id, status)
    request = DescribeInstancesRequest()
    request.set_InstanceIds(json.dumps([instance_id]))
    response = _send_request(request)
    instance_detail = None
    if response is not None:
        instance_list = response.get('Instances').get('Instance')
        for item in instance_list:
            if item.get('Status') == status:
                instance_detail = item
                break
        return instance_detail

# send open api request
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)

if __name__ == '__main__':
    logging.info("Create ECS by OpenApi!")
    create_instance_action()
    # create_prepay_instance_action()
```

Create a pay-as-you-go ECS instance

Required parameters:

- SecurityGroupId: the ID of the security group. A security group is similar to a firewall and uses security group rules to control inbound and outbound traffic of ECS instances in the security group. We recommend that you configure security group rules based on your actual needs. For more information, see CreateSecurityGroup.
- InstanceType: the instance type. For example, if you want to use the g6.large instance type with two vCPUs and 8 GiB memory, set this input parameter to *ecs.g6.large*. For more information, see Instance families.
- ImageId: the ID of the image. You can use a public or custom image. For more information, see DescribeImages.
- VSwitchId: the ID of the VSwitch. You must specify a VSwitch ID if you want to create an ECS instance of the VPC type. For more information, see DescribeVSwitches.

1. Create an ECS instance. The following code shows how to create a VPC-type ECS instance. The *cloud_ssd* value indicates that the instance uses a standard SSD as the system disk. The *optimized* value indicates that the instance is I/O optimized. Other request parameters are also supported in CreateInstance. For more information, see CreateInstance.

```
# create one after pay ecs instance.
def create_after_pay_instance(image_id, instance_type, security_group_id, vsw_vswitch_id):
    request = CreateInstanceRequest()
    request.set_ImageId(image_id)
    request.set_SecurityGroupId(security_group_id)
    request.set_InstanceType(instance_type)
    request.set_IoOptimized('optimized')
    request.set_VSwitchId(vsw_vswitch_id)
    request.set_SystemDiskCategory('cloud_ssd')
    response = _send_request(request)
    instance_id = response.get('InstanceId')
    logging.info("instance %s created task submit successfully.", instance_id)
    return instance_id
```

2. Query the status of the ECS instance. For more information about the states of ECS instances, see ECS instance lifecycle. Note the following items:
   - The StartInstance operation can be performed only on instances in the *Stopped* state.
   - The StopInstance operation can be performed only on instances in the *Running* state.

   You can query the status of a specific ECS instance by specifying InstanceId. When you call the DescribeInstances operation, you can pass in a JSON array of strings to query the instance status. For more information, see DescribeInstances. The following code queries the instance status and returns information about the instance:

```
# output the instance owned in current region.
def get_instance_detail_by_id(instance_id, status='Stopped'):
    logging.info("Check instance %s status is %s", instance_id, status)
    request = DescribeInstancesRequest()
    request.set_InstanceIds(json.dumps([instance_id]))
    response = _send_request(request)
    instance_detail = None
    if response is not None:
        instance_list = response.get('Instances').get('Instance')
        for item in instance_list:
            if item.get('Status') == status:
                instance_detail = item
                break
        return instance_detail
```

3. Start the ECS instance. After the ECS instance is created, the instance is in the *Stopped* state by default. If you want to put the instance into the *Running* state, call StartInstance. For more information, see StartInstance.

```
def start_instance(instance_id):
    request = StartInstanceRequest()
    request.set_InstanceId(instance_id)
    _send_request(request)
```

4. (Optional)Stop the ECS instance. If you want to change a running instance to the *Stopped* state, call StopInstance. For more information, see StopInstance.

```
def stop_instance(instance_id):
    request = StopInstanceRequest()
    request.set_InstanceId(instance_id)
    _send_request(request)
```

Check and start an ECS instance after creation

The operations of starting and stopping an ECS instance are asynchronous. You can use scripts to create an ECS instance, check its status, and perform the required operation. The sample code has the following logic:

1. Check whether the instance is in the *Stopped* state after the ID of the instance is obtained.
2. If the instance is in the *Stopped* state, call StartInstance.
3. Wait for the ECS instance to enter the *Running* state.

```
def check_instance_running(instance_id):
    detail = get_instance_detail_by_id(instance_id, INSTANCE_RUNNING)
    index = 0
    while detail is None and index < 60:
        detail = get_instance_detail_by_id(instance_id)
        time.sleep(10)
    if detail and detail.get('Status') == 'Stopped':
        logging.info("instance %s is stopped now.")
        start_instance(instance_id)
        logging.info("start instance %s job submit.")
    detail = get_instance_detail_by_id(instance_id, INSTANCE_RUNNING)
    while detail is None and index < 60:
        detail = get_instance_detail_by_id(instance_id, INSTANCE_RUNNING)
        time.sleep(10)
    logging.info("instance %s is running now.", instance_id)
    return instance_id
```

Assign a public IP address to an ECS instance

If you specify the public bandwidth when you create an ECS instance, you must call a specific API operation to assign a public IP address to the instance for Internet access. For more information, see AllocatePublicIpAddress

Create a subscription ECS instance

Prerequisites: The process of creating a subscription ECS instance by calling API operations is different from that in the ECS console. Fees are automatically deducted when you call API operations. The balance or credit of your account is sufficient. Fees are automatically deducted during the instance creation.

You must specify the billing method and duration. The duration is set to one month in the following sample code:

```
request.set_Period(1)
request.set_InstanceChargeType('PrePaid')
```

The following sample code shows how to create a subscription instance:

```
# create one prepay ecs instance.
def create_prepay_instance(image_id, instance_type, security_group_id, vsw_vswitch_id):
    request = CreateInstanceRequest()
    request.set_ImageId(image_id)
    request.set_SecurityGroupId(security_group_id)
    request.set_InstanceType(instance_type)
    request.set_IoOptimized('optimized')
    request.set_VSwitchId(vsw_vswitch_id)
    request.set_SystemDiskCategory('cloud_ssd')
    request.set_Period(1)
    request.set_InstanceChargeType('PrePaid')
    response = _send_request(request)
    instance_id = response.get('InstanceId')
    logging.info("instance %s created task submit successfully.", instance_id)
    return instance_id
```

3.3. Query an ECS instance

You can manage an ECS instance by using the ECS console or API operations. Alibaba Cloud provides SDKs in multiple languages to encapsulate APIs. This topic describes how to use an API operation to query an ECS instance in Python.

Context

For more information about the sample code of querying an ECS instance, see Sample code. For more information about the explanation of the sample code, see the following topics:

- Query the list of regions supported by the current account
- Query ECS instances in a specific region

Create a RAM user and obtain its AccessKey pair

An AccessKey pair that consists of an AccessKey ID and an AccessKey secret is required if you want to use API operations to manage an ECS instance. To ensure the security of cloud services, you must create a RAM user, generate an AccessKey pair for the RAM user, and authorize the RAM user to manage ECS instances by using APIs.

Perform the following operations to create a RAM user and obtain its AccessKey pair:

1. Create a RAM user. For more information, see Create a RAM user.
2. Obtain the AccessKey pair of the RAM user.
3. Authorize the RAM user to manage ECS instances. For more information, see Grant permissions to a RAM user.

Install ECS SDK for Python

Make sure that you have prepared the runtime environment for Python. Python 2.7 or later is used in this topic.

Run the following command to install SDK for Python:

```
pip install aliyun-python-sdk-ecs
```

If you are prompted that you do not have the permission, use the following **sudo** command to continue the installation:

```
sudo pip install aliyun-python-sdk-ecs
```

SDK for Python 2.1.2 is used in this topic.

Sample code

The following content is the sample code. You can modify the code and change the values of parameters based on your actual needs.

```
#  coding=utf-8
# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is greater than 2.1.2, you can use command 'pip show aliyun-python-sdk-e
cs' to check
import json
import logging
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.DescribeRegionsRequest import DescribeRegionsRequest
# configuration the log output formatter, if you want to save the output to file,
# append ",filename='ecs_invoke.log'" after datefmt.

logging.basicConfig(level=logging.INFO, format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelna
me)s %(message)s', datefmt='%a %d %b %Y %H:%M:%S')
clt = client.AcsClient('<accessKeyId>', '<accessSecret>', '<region-Id>')

# sample api to list aliyun open api.
def hello_aliyun_regions():
    request = DescribeRegionsRequest()
    response = _send_request(request)
    if response is not None:
        region_list = response.get('Regions').get('Region')
        assert response is not None
        assert region_list is not None
        result = map(_print_region_id, region_list)
        logging.info("region list: %s", result)
```

```
# output the instance owned in current region.
def list_instances():
    request = DescribeInstancesRequest()
    response = _send_request(request)
    if response is not None:
        instance_list = response.get('Instances').get('Instance')
        result = map(_print_instance_id, instance_list)
        logging.info("current region include instance %s", result)

def _print_instance_id(item):
    instance_id = item.get('InstanceId')
    return instance_id

def _print_region_id(item):
    region_id = item.get("RegionId")
    return region_id

# send open api request
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)

if __name__ == '__main__':
    logging.info("Hello Aliyun OpenApi!")
    hello_aliyun_regions()
    list_instances()
```

Query the list of regions supported by the current account

Create a file named *hello_ecs_api.py*. Before you use SDK for Python, use the Alibaba Cloud AccessKey ID and AccessKey secret of the RAM user to instantiate an AcsClient object.

> **Note** The AccessKey ID and AccessKey secret allow the RAM user to access ECS API of Alibaba Cloud with full permissions. You must keep the AccessKey pair confidential.

```
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.DescribeRegionsRequest import DescribeRegionsRequest
clt = client.AcsClient('<accessKeyId>', '<accessSecret>', '<region-Id>')
```

Query the list of regions that are supported by your current account after the AcsClient object is instantiated. For more information, see DescribeRegions.

```
def hello_aliyun_regions():
    request = DescribeRegionsRequest()
    response = _send_request(request)
    region_list = response.get('Regions').get('Region')
    assert response is not None
    assert region_list is not None
    result = map(_print_region_id, region_list)
    logging.info("region list: %s", result)
def _print_region_id(item):
    region_id = item.get("RegionId")
    return region_id
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)
hello_aliyun_regions()
```

Run `python hello_ecs_api.py` on the command line to obtain the list of regions that are supported by the current account. The following example shows a command output:

```
[u'cn-shenzhen', u'ap-southeast-1', u'cn-qingdao', u'cn-beijing', u'cn-shanghai', u'us-east-1', u'cn-hongk
ong', u'me-east-1', u'ap-southeast-2', u'cn-hangzhou', u'eu-central-1', u'ap-northeast-1', u'us-west-1']
```

Query ECS instances in a specific region

The process of querying the instance list is similar to that of querying the region list. You must replace the input parameter with DescribeInstancesRequest and configure relevant variables. For more information, see DescribeInstances.

```
def list_instances():
    request = DescribeInstancesRequest()
    response = _send_request(request)
    if response is not None:
        instance_list = response.get('Instances').get('Instance')
        result = map(_print_instance_id, instance_list)
        logging.info("current region include instance %s", result)
def _print_instance_id(item):
    instance_id = item.get('InstanceId')
    return instance_id
```

The following example shows a command output:

```
current region include instance [u'i-bp165p6xk2tmdhj0****', u'i-bp1a01zbqmsun5odo****']
```

For more information about API operations for resource query, see List of API operations by function. For example, to query the list of disks, you can call the DescribeDisks operation by replacing the input parameter with DescribeDisksRequest and configuring relevant variables.

3.4. Release an ECS instance

ECS resources can be created based on your needs. You can create custom resources during peak hours based on your needs and release these resources during off-peak hours. This topic describes how to use SDK for Python to release an ECS instance.

Context

After an ECS instance is released, the physical resources used by the instance will be recycled, such as disks and snapshots. All related data will be deleted and can never be recovered. If you want to retain the data, we recommend that you create snapshots of the disks before you release the ECS instance. The snapshots can be used to create a new ECS instance.

This topic provides the sample code to show how to release an ECS instance and also the explanation of the sample code. For more information, see the following topics:

- Stop an ECS instance
- Release an ECS instance
- Set the automatic release time for an ECS instance
- Cancel the automatic release

Sample code

The sample code is as follows.

> Note Exercise caution when you release an ECS instance.

```
# coding=utf-8
rr# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
```

```
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 2.1.2, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DeleteInstanceRequest import DeleteInstanceRequest
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.ModifyInstanceAutoReleaseTimeRequest import \
    ModifyInstanceAutoReleaseTimeRequest
from aliyunsdkecs.request.v20140526.StopInstanceRequest import StopInstanceRequest

# configuration the log output formatter, if you want to save the output to file,
# append ",filename='ecs_invoke.log'" after datefmt.

logging.basicConfig(level=logging.INFO, format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelna
me)s %(message)s',
                    datefmt='%a, %d %b %Y %H:%M:%S')
clt = client.AcsClient('<accessKeyId>', '<accessSecret>', '<region-Id>')


def stop_instance(instance_id, force_stop=False):
    '''
    stop one ecs instance.
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :param force_stop: if force stop is true, it will force stop the server and not ensure the data
    write to disk correctly.
    :return:
    '''
    request = StopInstanceRequest()
    request.set_InstanceId(instance_id)
    request.set_ForceStop(force_stop)
    logging.info("Stop %s command submit successfully.", instance_id)
    _send_request(request)


def describe_instance_detail(instance_id):
    '''
    describe instance detail
    :param instance_id: instance id of the ecs instance, like 'i-***'.
```

```
    :return:
    '''
    request = DescribeInstancesRequest()
    request.set_InstanceIds(json.dumps([instance_id]))
    response = _send_request(request)
    if response is not None:
        instance_list = response.get('Instances').get('Instance')
        if len(instance_list) > 0:
            return instance_list[0]


def check_auto_release_time_ready(instance_id):
    detail = describe_instance_detail(instance_id=instance_id)
    if detail is not None:
        release_time = detail.get('AutoReleaseTime')
        return release_time


def release_instance(instance_id, force=False):
    '''
    delete instance according instance id, only support after pay instance.
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :param force:
    if force is false, you need to make the ecs instance stopped, you can
    execute the delete action.
    If force is true, you can delete the instance even the instance is running.
    :return:
    '''
    request = DeleteInstanceRequest();
    request.set_InstanceId(instance_id)
    request.set_Force(force)
    _send_request(request)


def set_instance_auto_release_time(instance_id, time_to_release=None):
    '''
    setting instance auto delete time
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :param time_to_release: if the property is setting, such as '2017-01-30T00:00:00Z'
    it means setting the instance to be release at that time.
    if the property is None, it means cancel the auto delete time.
```

```
    :return:
    '''
    request = ModifyInstanceAutoReleaseTimeRequest()
    request.set_InstanceId(instance_id)
    if time_to_release is not None:
        request.set_AutoReleaseTime(time_to_release)
    _send_request(request)
    release_time = check_auto_release_time_ready(instance_id)
    logging.info("Check instance %s auto release time setting is %s. ", instance_id, release_time)


def _send_request(request):
    '''
    send open api request
    :param request:
    :return:
    '''
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)


if __name__ == '__main__':
    logging.info("Release ecs instance by Aliyun OpenApi!")
    stop_instance('i-bp1aet7s13lfpjop****')
    # set_instance_auto_release_time('i-bp1187lghfcy8nnz****', '2020-04-17T06:00:00Z')
    # set_instance_auto_release_time('i-bp1aet7s13lfpjop****')
    # release_instance('i-bp1aet7s13lfpjop****')
    # release_instance('i-bp1aet7s13lfpjop****', True)
```

Stop an ECS instance

Before you release an ECS instance, make sure that the ECS instance is in the Stopped state. After the ECS instance is stopped, you can restart the ECS instance based on your needs.

The command used to stop the ECS instance is easy to use. You can use the same command to stop pay-as-you-go and subscription instances. When the ForceStop parameter is set to *true*, the ECS instance is stopped but its data is not necessarily written to the disk, which is similar to a power failure. Therefore, if you want to release an ECS instance, set ForceStop to *true*.

```
def stop_instance(instance_id, force_stop=False):
    '''
    stop one ecs instance.
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :param force_stop: if force stop is true, it will force stop the server and not ensure the data
    write to disk correctly.
    :return:
    '''
    request = StopInstanceRequest()
    request.set_InstanceId(instance_id)
    request.set_ForceStop(force_stop)
    logging.info("Stop %s command submit successfully.", instance_id)
    _send_request(request)
```

Release an ECS instance

If you release an ECS instance that is not in the Stopped state, the following error may occur:

```
{"RequestId":"3C6DEAB4-7207-411F-9A31-6ADE54C268BE","HostId":"ecs-cn-hangzhou.aliyuncs.com","Co
de":"IncorrectInstanceStatus","Message":"The current status of the resource does not support this op
eration."}
```

If an ECS instance is in the *Stopped* state, you can release it. The following parameters are used to release an ECS instance:

- InstanceId: the ID of the instance.
- force: specifies whether to forcibly release the instance. If this parameter is set to *true*, the instance is released forcibly even when the instance is not in the *Stopped* state. Exercise caution when you set this parameter. Release by mistake may affect your business.

The following content is a request to release an ECS instance:

```
def release_instance(instance_id, force=False):
    '''
    delete instance according instance id, only support after pay instance.
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :param force:
    if force is false, you need to make the ecs instance stopped, you can
    execute the delete action.
    If force is true, you can delete the instance even the instance is running.
    :return:
    '''
    request = DeleteInstanceRequest();
    request.set_InstanceId(instance_id)
    request.set_Force(force)
    _send_request(request)
```

The following response is returned if the ECS instance is released:

```
{"RequestId":"689E5813-D150-4664-AF6F-2A27BB4986A3"}
```

Set the automatic release time for an ECS instance

You can set the automatic release time for an ECS instance to simplify instance management. When the release time is reached, Alibaba Cloud automatically releases the ECS instance without the need for manual intervention.

> **Note** Specify the automatic release time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC. If the value of the seconds place is not 00, the time will automatically be set to the start of the specified minute. The automatic release time ranges from 30 minutes later than the current time to three years later than the current time.

```
def set_instance_auto_release_time(instance_id, time_to_release = None):
    '''
    setting instance auto delete time
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :param time_to_release: if the property is setting, such as '2017-01-30T00:00:00Z'
    it means setting the instance to be release at that time.
    if the property is None, it means cancel the auto delete time.
    :return:
    '''
    request = ModifyInstanceAutoReleaseTimeRequest()
    request.set_InstanceId(instance_id)
    if time_to_release is not None:
        request.set_AutoReleaseTime(time_to_release)
    _send_request(request)
```

Run `set_instance_auto_release_time('i-1111', '2017-01-30T00:00:00Z')` to set the automatic release time.

After the time is set, you can call DescribeInstances to query the automatic release time:

```
def describe_instance_detail(instance_id):
    '''
    describe instance detail
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :return:
    '''
    request = DescribeInstancesRequest()
    request.set_InstanceIds(json.dumps([instance_id]))
    response = _send_request(request)
    if response is not None:
        instance_list = response.get('Instances').get('Instance')
        if len(instance_list) > 0:
            return instance_list[0]
def check_auto_release_time_ready(instance_id):
    detail = describe_instance_detail(instance_id=instance_id)
    if detail is not None:
        release_time = detail.get('AutoReleaseTime')
        return release_time
```

Cancel the automatic release

If you want to cancel the automatic release due to your business changes, run the following command to set the automatic release time to null:

```
set_instance_auto_release_time('i-1111')
```

Related information

- DeleteInstance
- ModifyInstanceAutoReleaseTime
- StopInstance
- DescribeInstances

3.5. Renew an ECS instance

You can renew an ECS instance in the ECS console or on the instance buy page. You can also call an API operation provided by Alibaba Cloud to renew an ECS instance or query the expiration time of the instance.

Context

The lifecycle is important to subscription ECS instances. If you fail to renew an ECS instance in a timely manner, the instance may be locked or even released. This can affect your service continuity. You can use API operations to query the expiration time of the ECS instance, and renew your instance.

This topic provides the sample code to show how to renew an ECS instance and also the explanation of the sample code. For more information, see the following topics:

- Query instances that will expire within the specified time range
- Renew an ECS instance
- Enable auto-renewal for ECS instances

Sample code

```
#  coding=utf-8
# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 4.4.3, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstanceAutoRenewAttributeRequest import \
    DescribeInstanceAutoRenewAttributeRequest
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.ModifyInstanceAutoRenewAttributeRequest import \
    ModifyInstanceAutoRenewAttributeRequest
from aliyunsdkecs.request.v20140526.RenewInstanceRequest import RenewInstanceRequest
```

```
logging.basicConfig(level=logging.INFO, format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelna
me)s %(message)s', datefmt='%a, %d %b %Y %H:%M:%S')
clt = client.AcsClient('<accessKeyId>', '<accessSecret>', '<region-Id>')

INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING = '2017-01-22T00:00Z'
INSTANCE_EXPIRE_END_TIME_IN_UTC_STRING = '2017-01-28T00:00Z'
def renew_job(page_size=100, page_number=1, check_need_renew=True, security_group_id=None):
    response = describe_need_renew_instance(page_size=page_size, page_number=page_number, che
ck_need_renew=check_need_renew, security_group_id=security_group_id)
    response_list = response.get('Instances').get('Instance')
    logging.info("%s instances need to renew", str(response.get('TotalCount')))
    if response_list > 0:
        instance_ids = ''
        for item in response_list:
            instance_id = item.get('InstanceId')
            instance_ids += instance_id + ','
            renew_instance(instance_id=instance_id)
        logging.info("%s execute renew action ready", instance_ids)

def describe_need_renew_instance(page_size=100, page_number=1, instance_id=None, check_need_re
new=True, security_group_id=None):
    request = DescribeInstancesRequest()
    if check_need_renew is True:
        request.set_Filter3Key("ExpiredStartTime")
        request.set_Filter3Value(INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING)
        request.set_Filter4Key("ExpiredEndTime")
        request.set_Filter4Value(INSTANCE_EXPIRE_END_TIME_IN_UTC_STRING)
    if instance_id is not None:
        request.set_InstanceIds(json.dumps([instance_id]))
    if security_group_id:
        request.set_SecurityGroupId(security_group_id)
    request.set_PageNumber(page_number)
    request.set_PageSize(page_size)
    return _send_request(request)

def describe_instance_auto_renew_setting(instance_ids, expected_auto_renew=True):
    describe_request = DescribeInstanceAutoRenewAttributeRequest()
    describe_request.set_InstanceId(instance_ids)
    response_detail = _send_request(request=describe_request)
    failed_instance_ids = ''
    if response_detail is not None:
```

```
    if response_detail is not None:
        attributes = response_detail.get('InstanceRenewAttributes').get('InstanceRenewAttribute')
        if attributes:
            for item in attributes:
                auto_renew_status = item.get('AutoRenewEnabled')
                if auto_renew_status ! = expected_auto_renew:
                    failed_instance_ids += item.get('InstanceId') + ','
    if len(failed_instance_ids) > 0:
        logging.error("instance %s auto renew not match expect %s.", failed_instance_ids,
                      expected_auto_renew)

def setting_instance_auto_renew(instance_ids, auto_renew=True):
    logging.info('execute enable auto renew ' + instance_ids)
    request = ModifyInstanceAutoRenewAttributeRequest();
    request.set_Duration(1);
    request.set_AutoRenew(auto_renew);
    request.set_InstanceId(instance_ids)
    _send_request(request)
    describe_instance_auto_renew_setting(instance_ids, auto_renew)

def check_instance_need_renew(instance_id):
    response = describe_need_renew_instance(instance_id=instance_id)
    if response is not None:
        return response.get('TotalCount') == 1
    return False

def renew_instance(instance_id, period='1'):
    need_renew = check_instance_need_renew(instance_id)
    if need_renew:
        _renew_instance_action(instance_id, period)
        # describe_need_renew_instance(instance_id=instance_id, check_need_renew=False)

def _renew_instance_action(instance_id, period='1'):
    request = RenewInstanceRequest()
    request.set_Period(period)
    request.set_InstanceId(instance_id)
    response = _send_request(request)
    logging.info('renew %s ready, output is %s ', instance_id, response)

def _send_request(request):
    request.set_accept_format('json')
```

```
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)


if __name__ == '__main__':
    logging.info("Renew ECS Instance by OpenApi!")
    # Query instances that will expire within the specified time range.
    describe_need_renew_instance()
    # Renew the instances and renewal fees are automatically deducted.
    # renew_instance('i-bp1aet7s13lfpjop****')
    # Query the auto-renewal status of the instances.
    # describe_instance_auto_renew_setting('i-bp1aet7s13lfpjop****,i-bp13uh1twnfv7vp8****')
    # Set auto-renewal for the instances.
    # setting_instance_auto_renew('i-bp1aet7s13lfpjop****,i-bp13uh1twnfv7vp8****')
```

Query instances that will expire within the specified time range

You can call the DescribeInstances operation to query instances that will expire within the time range specified by the ExpiredStartTime and ExpiredEndTime parameters. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC. If you want to filter instances by security group, add the ID of the security group.

```
INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING = '2017-01-22T00:00Z'
INSTANCE_EXPIRE_END_TIME_IN_UTC_STRING = '2017-01-28T00:00Z'


def renew_job(page_size=100, page_number=1, check_need_renew=True, security_group_id=None):
    response = describe_need_renew_instance(page_size=page_size, page_number=page_number, che
ck_need_renew=check_need_renew, security_group_id=security_group_id)
    response_list = response.get('Instances').get('Instance')
    logging.info("%s instances need to renew", str(response.get('TotalCount')))
    if response_list > 0:
        instance_ids = ''
        for item in response_list:
            instance_id = item.get('InstanceId')
            instance_ids += instance_id + ','
            renew_instance(instance_id=instance_id)
        logging.info("%s execute renew action ready", instance_ids)
```

Renew an ECS instance

You can renew only subscription ECS instances. Pay-as-you-go instances cannot be renewed. You can pay the renewal fee by account balance or credit. Fee deduction and order creation are in sync with API calling. Ensure that your account has sufficient funds to support automatic deduction of fees.

```
def _renew_instance_action(instance_id, period='1'):
    request = RenewInstanceRequest()
    request.set_Period(period)
    request.set_InstanceId(instance_id)
    response = _send_request(request)
    logging.info('renew %s ready, output is %s ', instance_id, response)
```

Fees are automatically deducted when the instance is renewed. After the renewal is complete, you can query the expiration time of the instance based on InstanceId. The expiration time may be updated in 10 seconds because the API operation is called asynchronously.

Enable auto-renewal for ECS instances

Alibaba Cloud provides auto-renewal for subscription ECS instances to help you reduce maintenance costs. Fees are automatically deducted at 08:00:00 nine days before the expiration date. If fee deduction fails on the first day, the deduction process will repeat on the following days until fees are deducted or resources are locked after nine days. You must make sure that your account has sufficient balance or credit.

- Query the auto-renewal status

  You can use an API operation to query and set the auto-renewal status. The API operation supports only subscription ECS instances. If you use the API operation on pay-as-you-go instances, an error is returned. You can query the auto-renewal status of up to 100 subscription ECS instances at a time. Separate multiple instance IDs with commas.

  The input parameter of DescribeInstanceAutoRenewAttribute is the instance ID.

  InstanceId: the IDs of instances. You can specify up to 100 subscription instance IDs at a time. Separate multiple instance IDs with commas.

```
def describe_instance_auto_renew_setting(instance_ids, expected_auto_renew=True):
    describe_request = DescribeInstanceAutoRenewAttributeRequest()
    describe_request.set_InstanceId(instance_ids)
    response_detail = _send_request(request=describe_request)
    failed_instance_ids = ''
    if response_detail is not None:
        attributes = response_detail.get('InstanceRenewAttributes').get('InstanceRenewAttribute')
        if attributes:
            for item in attributes:
                auto_renew_status = item.get('AutoRenewEnabled')
                if auto_renew_status ! = expected_auto_renew:
                    failed_instance_ids += item.get('InstanceId') + ','

describe_instance_auto_renew_setting('i-bp1aet7s13lfpjop****,i-bp13uh1twnfv7vp8****')
```

Sample responses:

```
{"InstanceRenewAttributes":{"InstanceRenewAttribute":[{"Duration":0,"InstanceId":"i-1111","AutoRe
newEnabled":false},{"Duration":0,"InstanceId":"i-2222","AutoRenewEnabled":false}]},"RequestId":"71
FBB7A5-C793-4A0D-B17E-D6Bxxxxxxxxx"}
```

If auto-renewal is set, the returned attribute of AutoRenewEnabled is *true*. Otherwise, the returned attribute is *false*.

- Enable and disable auto-renewal for ECS instances

  You must set the following input parameters to enable auto-renewal:

  - InstanceId: the IDs of instances. You can specify up to 100 subscription instance IDs at a time. Separate multiple instance IDs with commas.
  - Duration: the duration of ECS instances. Valid values: 1, 2, 3, 6, and 12. Unit: months.
  - AutoRenew: specifies whether to enable the auto-renewal feature. Valid values: *true* and *false*. Set the value to *true* to enable auto-renewal. Set the value to *false* to disable auto-renewal.

  ```
  def setting_instance_auto_renew(instance_ids, auto_renew = True):
      logging.info('execute enable auto renew ' + instance_ids)
      request = ModifyInstanceAutoRenewAttributeRequest();
      request.set_Duration(1);
      request.set_AutoRenew(auto_renew);
      request.set_InstanceId(instance_ids)
      _send_request(request)
  ```

  If the operation is successful, the following response is returned:

  ```
  {"RequestId":"7DAC9984-AAB4-43EF-8FC7-7D7xxxxxxxxx"}
  ```

After auto-renewal is enabled, you can query the auto-renewal status again. The system returns the auto-renewal duration and status in the following code:

{"InstanceRenewAttributes":{"InstanceRenewAttribute":[{"Duration":1,"InstanceId":"i-1111","AutoRenewEnabled":true},{"Duration":1,"InstanceId":"i-2222","AutoRenewEnabled":true}]},"RequestId":"7F4D14B0-D0D2-48C7-B310-B1DF713D4331"}

Related information

- DescribeInstances
- RenewInstance
- DescribeInstanceAutoRenewAttribute
- ModifyInstanceAutoRenewAttribute

3.6. Query available resources for configuration changes

This topic describes how to use Alibaba Cloud ECS SDK for Python to query available resources in Python 2.7 that you can use to change the configurations of an ECS instance.

Prerequisites

The AccessKey pair and the ID of the region where the resources reside are obtained. For more information, see and .

Context

You can call the ModifyInstanceSpec or ModifyPrepayInstanceSpec operation to change the configurations of an ECS instance. Before you call an API operation to change the configurations of an ECS instance, you can call the DescribeResourcesModification operation to query the following information:

- Available resources that you can use to upgrade the configurations of the ECS instance
- Available resources that you can use to upgrade the system disk

When you call the DescribeResourcesModification operation, you can set the `MigrateAcrossZone` parameter to true to query available resources that you can use to change the configurations of an ECS instance across generations. The private IP address of the ECS instance of the classic network type will change if you change the configurations of the instance across generations. If you are using a phased-out instance type, exercise caution when you upgrade a non-I/O optimized instance to an I/O optimized instance. The following table describes the results that may occur based on the network type and instance family if the configurations of an ECS instance are changed .

Item subject to change	Change configurations of a non-I/O optimized generation 1 instance across instance types		Change configurations of an instance of other generations	
	Classic network	VPC	Classic network	VPC
Private IP address	Changed	Unchanged	Changed	

Item subject to change		Change configurations of a non-I/O optimized generation 1 instance across instance types		Change configurations of an instance of other generations	
		Classic network	VPC	Classic network	VPC
Driver name (only in Linux)	Basic disk: *cloud*	Changed to xvda or xvdb			
	Ultra disk: *cloud_efficiency*	Changed to vda or vdb		Unchanged	
Software authorization code		Changed			

Install SDK for Python

The following example shows how to install SDK for Python in Linux:

- For root users, run the following command:

```
pip install aliyun-python-sdk-ecs
```

- For non-root users, run the following command:

```
sudo pip install aliyun-python-sdk-ecs
```

Usage scenarios

```
#  coding=utf-8

# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 4.6.3, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging

from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeResourcesModificationRequest import DescribeResourc
esModificationRequest

logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',
                    datefmt='%a, %d %b %Y %H:%M:%S')

# Your AccessKey ID.
```

```
ak_id = "YOU_ACCESS_KEY_ID"

# Your AccessKey secret.
ak_secret = "YOU_ACCESS_SECRET"

# The region ID of the ECS resource.
region_id = "cn-hangzhou"

# The resource ID, such as the instance ID.
resource_id = "i-bp1187lghfcy8nnz****"

# The target resource type. Valid values: InstanceType and SystemDisk.
destination_instance_type ='InstanceType'
destination_system_disk ='SystemDisk'

# The operation that you want to perform to change configurations. Valid values: Upgrade, Downgrad
e, RenewDowngrade, and RenewModify.
operation_type = 'Upgrade'

# The target instance type. You can call DescribeInstanceTypes to query the most recent list of instan
ce types. When the DestinationResource parameter is set to SystemDisk, you must specify the Instanc
eType parameter.
instance_type = "ecs.g6.large"

"""
Exercise caution when you set the MigrateAcrossZone parameter. MigrateAcrossZone (Boolean) speci
fies whether the configurations of an ECS instance can be changed across generations. Default value:
False.
When the MigrateAcrossZone parameter is set to True, and you upgrade the configurations of the ECS
instance based on the returned information, take note of the following considerations:
Instances of the classic network type:
1. For generation 1 instances, when non-I/O optimized instances are upgraded to I/O optimized instan
ces, the information of the instances is changed, such as the private IP addresses, drive names, and s
oftware authorization codes. For Linux instances, basic disks (cloud) are identified as xvda or xvdb. Ul
tra disks (cloud_efficiency) and standard SSDs (cloud_ssd) are identified as vda or vdb.
2. For instances of other generations, the private IP addresses of the instances are changed when yo
u change the configurations of the instances across generations.
Instances of the VPC type: For generation 1 instances, when non-I/O optimized instances are upgrade
d to I/O optimized instances, the information of the instances is changed, such as the drive names and
software authorization codes. For Linux instances, basic disks (cloud) are identified as xvda or xvdb. U
ltra disks (cloud_efficiency) and standard SSDs (cloud_ssd) are identified as vda or vdb.
```

```
"""
migrate_across_zone = False

clt = client.AcsClient(ak_id, ak_secret, region_id)

# Query available resources that can be used to upgrade instance types.
def describe_resource_instance_type():
    request = build_request()
    request.set_DestinationResource(destination_instance_type)
    _execute_request(request)

# Query available resources that can be used to upgrade system disks.
def describe_resource_system_disk():
    request = build_request()
    request.set_DestinationResource(destination_system_disk)
    request.set_InstanceType(instance_type)
    _execute_request(request)

def _execute_request(request):
    response = _send_request(request)
    if response is None:
        print ('response is None')
        return
    if response.get('Code') is None:
        availableZones = response.get('AvailableZones').get('AvailableZone')
        if availableZones is None:
            print ('availableZones is None')
            return
        for availableZone in availableZones:
            zoneId = availableZone.get('ZoneId')
            values = []
            availableResources = availableZone.get('AvailableResources').get('AvailableResource')
            if availableResources is None:
                print ('availableResources is None')
                return
            for availableResource in availableResources:
                supportedResources = availableResource.get('SupportedResources').get('SupportedResource
')
                if supportedResources is None:
                    print ('supportedResource is None')
```

```
                return
            for supportedResource in supportedResources:
                status = supportedResource.get('Status')
                if status == "Available":
                    value = supportedResource.get('Value')
                    values.append(value)
        print ("ecs in zone %s resource value list is %s"%(zoneId, values))

def build_request():
    request = DescribeResourcesModificationRequest()
    request.set_ResourceId(resource_id)
    request.set_MigrateAcrossZone(migrate_across_zone)
    request.set_OperationType(operation_type)
    return request

# Initiate an API request.
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)
```

Scenario 1: Query available resources that can be used to upgrade instance types

```
# Query available resources that can be used to upgrade instance types.
def describe_resource_instance_type():
    request = build_request()
    request.set_DestinationResource(destination_instance_type)
    _execute_request(request)
```

Scenario 2: Query available resources that can be used to upgrade system disks

```
# Query available resources that can be used to upgrade system disks.
def describe_resource_system_disk():
    request = build_request()
    request.set_DestinationResource(destination_system_disk)
    request.set_InstanceType(instance_type)
    _execute_request(request)
```

Sample code

```
#  coding=utf-8

# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 4.6.3, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging

from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeResourcesModificationRequest import DescribeResourc
esModificationRequest

logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',
                    datefmt='%a, %d %b %Y %H:%M:%S')

# Your AccessKey ID.
ak_id = "YOU_ACCESS_KEY_ID"

# Your AccessKey secret.
ak_secret = "YOU_ACCESS_SECRET"

# The region ID of the ECS resource.
region_id = "cn-hangzhou"

# The resource ID, such as the instance ID.
resource_id = "i-bp1187lghfcy8nnz****"

# The target resource type. Valid values: InstanceType and SystemDisk.
destination_instance_type ='InstanceType'
destination_system_disk ='SystemDisk'
```

```
destination_system_disk = SystemDisk

# The operation that you want to perform to change configurations. Valid values: Upgrade, Downgrad
e, RenewDowngrade, and RenewModify.
operation_type = 'Upgrade'

# The target instance type. You can call DescribeInstanceTypes to query the most recent list of instan
ce types. When the DestinationResource parameter is set to SystemDisk, you must specify the Instanc
eType parameter.
instance_type = "ecs.g6.large"

"""
Exercise caution when you set the MigrateAcrossZone parameter. MigrateAcrossZone (Boolean) speci
fies whether the configurations of an ECS instance can be changed across generations. Default value:
False.
When the MigrateAcrossZone parameter is set to True, and you upgrade the configurations of the ECS
instance based on the returned information, take note of the following considerations:
Instances of the classic network type:
1. For generation 1 instances, when non-I/O optimized instances are upgraded to I/O optimized instan
ces, the information of the instances is changed, such as the private IP address, drive name, and soft
ware authorization code. For Linux instances, basic disks (cloud) are identified as xvda or xvdb. Ultra
disks (cloud_efficiency) and standard SSDs (cloud_ssd) are identified as vda or vdb.
2. For instances of other generations, the private IP addresses of the instances are changed when yo
u change the configurations of the instances across generations.
Instances of the VPC type: For generation 1 instances, when non-I/O optimized instances are upgrade
d to I/O optimized instances, the information of the instances is changed, such as the drive name and
software authorization code. For Linux instances, basic disks (cloud) are identified as xvda or xvdb. Ul
tra disks (cloud_efficiency) and standard SSDs (cloud_ssd) are identified as vda or vdb.
"""
migrate_across_zone = False

clt = client.AcsClient(ak_id, ak_secret, region_id)

# Query available resources that can be used to upgrade instance types.
def describe_resource_instance_type():
    request = build_request()
    request.set_DestinationResource(destination_instance_type)
    _execute_request(request)

# Query available resources that can be used to upgrade system disks.
def describe_resource_system_disk():
```

```
    request = build_request()
    request.set_DestinationResource(destination_system_disk)
    request.set_InstanceType(instance_type)
    _execute_request(request)

def _execute_request(request):
    response = _send_request(request)
    if response is None:
        print ('response is None')
        return
    if response.get('Code') is None:
        availableZones = response.get('AvailableZones').get('AvailableZone')
        if availableZones is None:
            print ('availableZones is None')
            return
        for availableZone in availableZones:
            zoneId = availableZone.get('ZoneId')
            values = []
            availableResources = availableZone.get('AvailableResources').get('AvailableResource')
            if availableResources is None:
                print ('availableResources is None')
                return
            for availableResource in availableResources:
                supportedResources = availableResource.get('SupportedResources').get('SupportedResource
')
                if supportedResources is None:
                    print ('supportedResource is None')
                    return
                for supportedResource in supportedResources:
                    status = supportedResource.get('Status')
                    if status == "Available":
                        value = supportedResource.get('Value')
                        values.append(value)
            print ("ecs in zone %s resource value list is %s"%(zoneId, values))

def build_request():
    request = DescribeResourcesModificationRequest()
    request.set_ResourceId(resource_id)
    request.set_MigrateAcrossZone(migrate_across_zone)
    request.set_OperationType(operation_type)
    return request
```

```
# Initiate an API request.
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)


if __name__ == '__main__':
    print ("hello ecs describe resources")
    # Query available resources that can be used to upgrade instance types.
    describe_resource_instance_type()
    # Query available resources that can be used to upgrade system disks.
    # describe_resource_system_disk()
```