Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the 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 contact Alibaba Cloud directly if you discover any errors in this document.
## Document conventions

<table>
<thead>
<tr>
<th>Style</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.</td>
<td>Danger: Resetting will result in the loss of user configuration data.</td>
</tr>
<tr>
<td>!</td>
<td>A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.</td>
<td>Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.</td>
</tr>
<tr>
<td>!</td>
<td>A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.</td>
<td>Notice: If the weight is set to 0, the server no longer receives new requests.</td>
</tr>
<tr>
<td>![list]</td>
<td>A note indicates supplemental instructions, best practices, tips, and other content.</td>
<td>Note: You can use Ctrl + A to select all files.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Closing angle brackets are used to indicate a multi-level menu cascade.</td>
<td>Click Settings &gt; Network &gt; Set network type.</td>
</tr>
<tr>
<td><strong>Bold</strong></td>
<td>Bold formatting is used for buttons, menus, page names, and other UI elements.</td>
<td>Click OK.</td>
</tr>
<tr>
<td><strong>Courier font</strong></td>
<td>Courier font is used for commands.</td>
<td>Run the cd /d C:/window command to enter the Windows system folder.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic formatting is used for parameters and variables.</td>
<td>bae log list --instanceid Instance_ID</td>
</tr>
<tr>
<td>[] or [a</td>
<td>b]</td>
<td>This format is used for an optional value, where only one item can be selected.</td>
</tr>
<tr>
<td>Style</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>{} or {a</td>
<td>b}</td>
<td>This format is used for a required value, where only one item can be selected.</td>
</tr>
</tbody>
</table>
Contents

Legal disclaimer........................................................................................................ I
Document conventions............................................................................................ I
1 SDK overview........................................................................................................ 1

2 Java example......................................................................................................... 2
  2.1 Install the Java SDK.......................................................................................... 2
  2.2 General process................................................................................................ 6
  2.3 Query available instance resources................................................................... 11
  2.4 Create an ECS instance.................................................................................... 16
    2.4.1 Batch create ECS instances..................................................................... 16
    2.4.2 Create a preemptible instance.................................................................. 19
  2.5 Query an ECS instance..................................................................................... 24
  2.6 Change the instance type of an ECS instance................................................... 26
  2.7 Release an ECS instance................................................................................. 28

3 Python example.................................................................................................... 30
  3.1 Create multiple instances at a time................................................................. 30
  3.2 Create an ECS instance.................................................................................... 34
  3.3 Manage instances............................................................................................ 40
  3.4 Release an instance.......................................................................................... 43
  3.5 Renew an instance............................................................................................ 48
  3.6 Query available resources for configuration changes....................................... 53
1 SDK overview

This topic describes the ECS SDKs in different programming language and lists the addresses for obtaining the latest SDK versions.

SDK introduction

The Alibaba Cloud ECS SDK encapsulates the API version released on May 6, 2014. It identifies callers based on their AccessKey pairs and provides functions such as automatic signature to allow you to create and manage resources by calling API operations.

To use ECS features, you must install both the ECS SDK and the Alibaba Cloud core library. For example, to use the Java SDK, you must download aliyun-java-sdk-core and aliyun-java-sdk-ecs.

Supported SDKs

The following table lists the SDKs supported by ECS. You can view the SDK update history, obtain the installation package, and view the help document in the GitHub repository.

<table>
<thead>
<tr>
<th>Programming language</th>
<th>GitHub address</th>
<th>GitHub document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>aliyun-openapi-java-sdk</td>
<td>README.md</td>
</tr>
<tr>
<td>Python</td>
<td>aliyun-openapi-python-sdk</td>
<td>README.md</td>
</tr>
<tr>
<td>PHP</td>
<td>aliyun-openapi-php-sdk</td>
<td>README.md</td>
</tr>
<tr>
<td>.NET</td>
<td>aliyun-openapi-net-sdk</td>
<td>README.md</td>
</tr>
<tr>
<td>C++</td>
<td>aliyun-openapi-cpp-sdk</td>
<td>README.md</td>
</tr>
<tr>
<td>Go</td>
<td>alibaba-cloud-sdk-go</td>
<td>README.md</td>
</tr>
<tr>
<td>Node.js</td>
<td>aliyun-openapi-nodejs-sdk</td>
<td>README.md</td>
</tr>
</tbody>
</table>

References

- The Alibaba Cloud Developer Center brings together popular SDKs for a variety of programming languages with information such as Quick Start Tutorials and open source projects. For more information, visit Alibaba Cloud Developer Center
- For more information about how to obtain an AccessKey pair, see #unique_4.
- For information about the endpoint list of ECS, see #unique_5.
2 Java example

2.1 Install the Java SDK

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

Prerequisites

JDK 1.6 is installed before you install the ECS Java SDK.

Context

You can download aliyun-java-sdk-ecs or use the SDK through Maven coordinates.

In this example, the client uses the Windows 7 64-bit operating system and the environment installed for Java is Eclipse Luna.

Procedure

1. Download the Maven software corresponding to the operating system from the official Maven website (Download Apache Maven).

   You can use the checksum file to verify whether the downloaded file is correct.

2. Find and decompress the Maven package. Add the following Maven library information to the settings.xml file in the conf folder.

```xml
<repositories>
  <repository>
    <id>sonatype-nexus-staging</id>
    <name>Sonatype Nexus Staging</name>
    <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>
```
3. Log on to the Alibaba Cloud Developer Center (https://developer.aliyun.com/tools/sdk#/java) and add the Maven library stored in the Alibaba Cloud SDK to the Maven software.
4. Create a Maven project using either of the following methods:

- Method 1: Add a Maven project in Eclipse.

- Method 2: Convert an existing project to a Maven project.
5. Add **dependency** to the Maven project using either of the following methods:

- **Method 1:** Open the pom.xml file under the Maven project and add dependencies including aliyun-java-sdk-core, aliyun-java-sdk-ecs, and fastjson. You can add different dependencies as needed.

```xml
<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>
```

- **Method 2:** Add selected dependencies from the graphical user interface.
6. Save the changes.

The system automatically downloads and adds the .jar Java SDK to the Maven dependencies.

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 #unique_4.

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 #unique_9.

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 #unique_10.
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.

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

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 #unique_11.

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

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

   Subsequently obtain `response` from IClientProfile.

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

3. Create a request for the API operation 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`.

   ```java
   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.

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

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

   ```java
   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 `getImageName()` or `getImageId`.

```java
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

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

   • Client error

   ```java
   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:

```json
{
    "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.

```java
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*****", "BNPO1zoNSi484oizGM9fzzwJ*****");
        IAcsClient client = new DefaultAcsClient(profile);

        DescribeImagesRequest request = new DescribeImagesRequest();
        request.setRegionId("cn-hangzhou");
        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();
        }
    }
}
```
### 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 #unique_14. 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.

```java
/**
 * 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, zoneld 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() {
    try {
        // Code for calling DescribeAvailableResource
    } catch (ClientException e) {
        System.out.println("ErrCode:" + e.getErrCode());
        System.out.println("ErrMsg:" + e.getErrMsg());
        System.out.println("RequestId:" + e.getRequestId());
    }
}
```
DescribeAvailableResourceRequest describe = new DescribeAvailableResourceRequest();
describe.setRegionId("cn-hangzhou");
describe.setInstanceChargeType("PrePaid");
// 1. Query available I/O optimized resources
describe.setDestinationResource(IOOPTIMIED);
Map<String, Set<String>> ioOptimeds = doActionAndProcessResponse(describe);
if (null == ioOptimeds) {
    return null;
}
// Set allTypes to specify a list of instance types, set the zone ID as the key, and enter
// the available instance types in the corresponding zone as the value.
Map<String, Set<String>> allTypes = new HashMap<String, Set<String>>();
for (String zoneId : ioOptimeds.keySet()) {
    describe.setZoneId(zoneId);
    describe.setDestinationResource(INSTANCETYPE);
    for (String iopts : ioOptimeds.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 PostPaid. 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>> doDescribeScene2() {
    DescribeAvailableResourceRequest describe = new DescribeAvailableResourceRequest();
    describe.setRegionId("cn-hangzhou");
    describe.setInstanceChargeType("PostPaid");
    // 1. Query available I/O optimized resources
    describe.setDestinationResource(IOOPTIMIED);
Map<String, Set<String>> ioOptimizados = doActionAndProcessResponse(describe);
   if (null == ioOptimizados) {
       return null;
   }
   // Set allTypes to specify a list of instance types, set the zone ID as the key, and enter
   the available instance types in the corresponding zone as the value.
   Map<String, Set<String>> allTypes = new HashMap<String, Set<String>>();
   for (String zoneId : ioOptimizados.keySet()) {
       describe.setZoneId(zoneId);
       describe.setDestinationResource(INSTANCETYPE);
       for (String iopts : ioOptimizados.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 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 PostPaid. Set the SpotStrategy parameter to SpotWithPriceLimit or
   SpotAsPriceGo.
   * 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>> doDescribeScene3()
    {
        DescribeAvailableResourceRequest describe = new DescribeAvailableResourceReque
st();
        describe.setRegionId("cn-hangzhou");
        describe.getInstanceChargeType("PostPaid");
        describe.setSpotStrategy("SpotWithPriceLimit");
        // describe.setSpotStrategy("SpotAsPriceGo");
        // 1. Query available I/O optimized resources
        describe.setDestinationResource(IOOPTIMIZED);
        Map<String, Set<String>> ioOptimizados = doActionAndProcessResponse(describe);
        if (null == ioOptimizados) {
            return null;
        }
// Set allTypes to specify a list of instance types, set the zone ID as the key, and enter
the available 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 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(IOOPTIMIED);
    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()) {
def 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.

```java
/**
 * Assume that you are querying the instance types that are available for VPC-type subscription instances 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 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:
 * 3. The list of qualified instance types that are available in the specified zone.
 * 4. 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(IOOPTIMIED);
    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 : ioOptimeds){
    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 topics
#unique_14

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
  #unique_17.
- 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-bp1fg655nh
  68xyz9i***. 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 #unique_18.

**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:

```java
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>", "<yourAccessSecret>");
        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 the disk.
List<RunInstancesRequest.DataDisk> dataDiskList = new ArrayList<RunInstancesRequest.DataDisk>();
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 will 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.getInstanceChargeType("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:

```json
{
"RequestId":"04F0F334-1335-436C-A1D7-6C044FE73368",
"InstanceIdSets":{
    "InstanceIdSet":[
        "i-instanceid1",
        "i-instanceid2",
        "i-instanceid3"
    ]
}
```
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 #unique_17.

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 #unique_24.
**Procedure**

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

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

   public ApiCaller() {
       profile = DefaultProfile.getProfile("cn-hangzhou", AKSUtil.accessKeyId, AKSUtil.
       accessKeySecret);
       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.

   ```java
   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 queried.
       request.setZoneId("cn-hangzhou-h");
       // If no instance type is specified, all instance types that can be selected to create preemptible 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:

```json
{
  "RequestId": "D8491D5E-AB8A-4E22-BDB4-EEE1F1C8241",
  "AvailableZones": [
    {
      "Status": "Available",
      "RegionId": "cn-hangzhou",
      "AvailableResources": [
        {
          "Type": "InstanceType",
          "SupportedResources": [
            {
              "Status": "Available",
              "Value": "ecs.g5.large",
              "StatusCategory": "WithStock"
            }
          ]
        }
      ],
      "ZoneId": "cn-hangzhou-b",
      "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.

```java
public class DescribeSpotPriceHistorySample {
    public static void main(String[] args) {
        ApiCaller caller = new ApiCaller();
        List<DescribeSpotPriceHistoryResponse.SpotPriceType> result = new ArrayList<DescribeSpotPriceHistoryResponse.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 period for which the historical prices of preemptible instances are to be queried. The time period is three days by default.
// request.setEndTime("2017-09-28T08:45:08Z"); // Optional. The end time of a time period for 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.getSpotPrice() + "-->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.

```java
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-bp11nhf94ivkdxb9****"); // The ID of the security group.
        request.setImageId("m-bp146shijn7hujkui9****"); // We recommend that you use a custom image to create instances more quickly.
        request.setVSwitchId("vsw-bp164cyonthfudn9jks****"); // The ID of the VSwitch that is in the same VPC as the security group.
        request.setInstanceType("ecs.g5.large"); // Enter the instance type you want to purchase after the query.
    }
}
```
Elastic Compute Service

SDK Sample / 2 Java example

```java
request.setSystemDiskCategory("cloud_ssd"); // The category of the system disk. Valid values: 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. Manually set the maximum price. SpotAsPriceGo: The system provides a price automatically based on the market price.
request.setSpotPriceLimit(0.25F); // If the bidding mode is set to SpotWithPriceLimit, the maximum 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 configured, 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.

a) 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.

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

```java
public class DescribeInstancesSample {
    public static void main(String[] args) throws InterruptedException {
        ApiCaller caller = new ApiCaller();
        JSONArray allInstances = new JSONArray();
        allInstances.addAll(Arrays.asList("i-bp18hgfai8ek0w0***", "i-bp1ecbyds24ij63w1****");
        while (! allInstances.isEmpty()) {
            DescribeInstancesRequest request = new DescribeInstancesRequest();
            request.setRegionId("cn-hangzhou");
            request.setInstanceIds(allInstances.toJSONString()); // Specify the ID of the instance and only query the instance whose ID has been specified.
            DescribeInstancesResponse response = caller.doAction(request);
```

Issue: 20200417
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.getZoneld());
        if (instance.getOperationLocks() != null) {
            for (DescribeInstancesResponse.Instance.LockReason lockReason : instance.getOperationLocks()) {
                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 instance from the instance ID list.
                    System.out.println("Preemptible instance will be recycled immediately, instance id: " + instance.getInstanceId());
                    allInstances.remove(instance.getInstanceId());
                }
            }
        }
    }
}
System.out.println("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 topics
#unique_19
#unique_25
#unique_14
#unique_20
#unique_26
#unique_27
#unique_18

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.

```java
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>", "<yourAccessSecret>");
        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 topics

#unique_25

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:

```java
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",
               "/yourAccessSecret/");
        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 downgrade 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:

```java
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 >", "</yourAccessSecret>");
        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 topics
#unique_30
#unique_31

## 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.

### Prerequisites

You must have at least one existing ECS instance. For more information, see [Batch create ECS instances](#unique_31).

### Sample code

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

```java
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.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>", "<yourAccessSecret>"."
        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 topics

#unique_33
#unique_34
3 Python example

3.1 Create multiple instances at a time

RunInstances can create multiple ECS instances at a time. It helps you fast develop and deploy application.

Compared with #unique_37, #unique_38 has the following benefits:

- RunInstances contains Amount to create and automatically run up to 100 instances or preemptible instances for one request.
- When an instance is created, the instance status automatically becomes Starting and then Running. You do not need to call the StartInstance operation.
- Instances have Internet IPs allocated if you set the value of InternetMaxBandwidthOut greater than 0.
- You can also create 100 preemptable instances at a time to fully meet your requirements.
- Release plan can be scheduled by setting AutoReleaseTime, and the number of created parameters can be set by configuring Amount. The error codes and available parameters of RunInstances are completely compatible with CreateInstance.
- Status polling of the created instances is allowed since InstanceIdSets lists all the InstanceIds after the request.

Prerequisites

Make sure you have created an AccessKey.

⚠️ Notice:

Do not use the AccessKey of the primary account. If it is disclosed, your resources may be unsafe. Use the AccessKey of an RAM user account to reduce the risk of AccessKey disclosure.

Install ECS Python SDK

Make sure that you have runtime for Python. In this document, we take a version later than Python 2.7
as an example and the **SDK version** is 4.4.3.

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

If you get any message indicating that you have no operation permission, switch to `sudo`.

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

The version of the SDK used in this article is 4.4.3. If you are using an older version of the SDK, update it.

**Create instances**

Create RunInstancesRequest object and then enter the related parameters:

In this example, we create two instances and specify to automatically check the instance status every 10 seconds. The creation procedure ends when the instance status turns into Running.

```
# your access key Id
ak_id = "YOU_ACCESS_KEY_ID"
# your access key secret
ak_secret = "YOU_ACCESS_SECRET"
region_id = "cn-beijing"
# your expected instance type
instance_type = "ecs.n4.small"
# The selected vswitchId
vswitch_id = "vws-xxxxx"
# The selected image info
image_id = "centos_7_03_64_20G_alibase_20170818.vhd"
# The selected security group of VPC network
security_group_id = "sg-xxxxx"
# instance number to launch, support 1-100, default value is 100
amount = 2;
# The auto release time is in accordance with ISO8601 and must be UTC. The format is `yyyy-MM-ddTHH:mm:ssZ`. The release time must be at least 30 minutes later than the current time and less than 3 years from the current time.
auto_release_time = "2017-12-05T22:40:00Z"
clt = client.AcsClient(ak_id, ak_secret, 'cn-beijing')
# create instance automatic running
def batch_create_instance():
    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_InstanceId(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("Instance12-04")
    request.set_InstanceType(instance_type)
    return request

# 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)

Create instances with Internet IP

Create instances and add a line of attribute to specify Internet bandwidth. In this example, we assign 1 Mbit/s bandwidth for each instance.

# create instance with public ip.
def batch_create_instance_with_public_ip():
    request = build_request()
    request.set_Amount(amount)
    request.set_InternetMaxBandwidthOut(1)
    _execute_request(request)

Create instances with auto release time

Create instances and add a line of attribute to specify auto release time for instances. The auto release time is in accordance with ISO8601 and must be UTC. The format is YYYY-MM-DDTHH:mm:ssZ. The release time cannot be 30 minutes earlier than the current time or more than 3 years from the current time.

# create instance with auto release time.
def batch_create_instance_with_auto_release_time():
    request = build_request()
    request.set_Amount(amount)
    request.set_AutoReleaseTime(auto_release_time)
    _execute_request(request)

Complete example code

The complete example code is as follows.

```python
#  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 %a, %d %b %Y %H:%M:%S'
)
# your access key Id
ak_id = "YOU_ACCESS_KEY_ID"
# your access key secret
ak_secret = "YOU_ACCESS_SECRET"
region_id = "cn-beijing"
# your expected instance type
instance_type = "ecs.n4.small"
# The selected vswwitchId.
vswwitch_id = "vws-xxxxx"
# The selected image info
image_id = "centos_7_03_64_20G_alibase_20170818.vhd"
# The selected security group of VPC network
security_group_id = "sg-xxxxx"
# instance number to launch, support 1-100, default value is 100
amount = 2;
# The auto release time is in accordance with ISO8601 and must be UTC. The format is `YYYY-MM-DDTHH:mm:ssZ`. The release time must be at least 30 minutes later than the current time and less than 3 years from the current time.
auto_release_time = "2017-12-05T22:40:00Z"
clt = client.AcsClient(ak_id, ak_secret, 'cn-beijing')
# create instance automatic running
def batch_create_instance():
    request = build_request()
    request.set_Amount(amount)
    _execute_request(request)
# create instance with public ip.
def batch_create_instance_with_public_ip():
    request = build_request()
    request.set_Amount(amount)
    request.set_InternetMaxBandwidthOut(1)
    _execute_request(request)
# create instance with auto release time.
def batch_create_instance_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("Instance12-04")
    request.set_InstanceType(instance_type)
    return request

# 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__':
    print "hello ecs batch create instance"
    # batch_create_instance()
    # batch_create_instance_with_public_ip()
    # batch_create_instance_with_auto_release_time()

### 3.2 Create an ECS instance

This topic describes how to create and manage an Alibaba Cloud ECS instance (either a Pay-As-You-Go or Subscription instance) by using Alibaba Cloud APIs.

The APIs described in this topic allow you to:

- **Create an ECS instance**
- **Query the status of an ECS instance**
- **Start an ECS instance**
- **Stop an ECS instance**
- **Allocate a public IP address**

**Create a Pay-As-You-Go ECS instance**

**Required parameters**

- **SecurityGroupId**: Security group ID. A security group is similar to a firewall and uses security group rules to control network access requests of instances. We recommend
that you configure access rules only according to the actual needs. For more information, see CreateSecurityGroup.

- **InstanceType**: Instance type. The option “one-core 2GiB n1.small” indicates that the input parameter is ecs.n1.small. For more information, see Instance type families.

- **ImageId**: Image ID. You can use public images or custom images. For more information, see DescribeImages.

**Step 1: Create an ECS instance**

The following code example shows the parameters optimized and cloud_ssd, which indicate the creation of an I/O optimized VPC instance with SSD as the system disk. For more information about other supported request parameters, see CreateInstance.

```python
# create one postpaid ecs instance.
def create_after_pay_instance(image_id, instance_type, security_group_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-vswitchid')
    request.set_VSwitchId('vsw-vswitchid')
    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;
```

After an ECS instance is created, the corresponding instance ID and request ID are returned as follows:

```json
{"InstanceId":"i-***","RequestId":"006C1303-BAC5-48E5-BCDF-7FD5C2E6395D"}
```

If the creation of an ECS instance fails, a corresponding error code and error message are returned as follows:

```json
```

**Step 2: Query ECS instance status**

Before you query the status of an ECS instance, note the following:

- The StartInstance operation can be performed on **Stopped** instances only.
- The StopInstance operation can be performed on **Running** instances only.
Note:
For more information about different ECS status, see ECS instance lifecycle.

To query the status of a specific ECS instance, you can specify its InstanceId. Specifically, when using DescribeInstances, you can pass the parameters in the form of a JSON array to query the instance status. The following code queries instance details:

```python
# 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;
```

Step 3: Start an ECS instance

After an ECS instance is created, the default instance status is Stopped. To change the instance to the Running status, call StartInstance:

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

(Optional) Step 4: Stop an ECS instance

To stop an ECS instance, call StopInstance:

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

Enable “ECS automatic startup” when creating an ECS instance

The ECS start and stop operations are asynchronous. You can use scripts to create an ECS instance, check its status, and perform the required operations. Specifically, the following code:

1. Checks whether the specified instance is in the Stopped status after obtaining an instance ID.
2. Calls StartInstance if it is in the Stopped status.
3. Waits for the instance to change to the **Running** status.

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

### Allocate a public IP address

If you specify the Internet bandwidth when you create an ECS instance, you need to call an API to allocate a public IP address to the instance for Internet access. For more information, see [AllocatePublicIpAddress](#).

### Create a Subscription ECS instance

Alibaba Cloud APIs also support the creation of Subscription ECS instances. If you want to create a Subscription ECS instance, make sure that you have a sufficient account balance or credit amount associated to your account so that the fees can be deducted directly during instance creation.

When you create a Subscription ECS instance, you only need to specify the payment option and duration. In the following code example, the billing duration of the instance is set to one month.

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

The complete code for the creation of a Subscription ECS instance is as follows:

```python
# create one prepay ecs instance.
def create_prepay_instance(image_id, instance_type, security_group_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_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;

Example

The following example uses Python to describe the creation of an ECS instance.

```python
# 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 2.1.2, 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] %(levelname)s %(message)s',
    datefmt='%a, %d %b %Y %H:%M:%S')
clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secret', 'cn-beijing')
IMAGE_ID = 'ubuntu1404_64_40G_cloudinit_20160727.raw'
INSTANCE_TYPE = 'ecs.s2.large'  # 2c4g generation 1
SECURITY_GROUP_ID = 'sg-****'
INSTANCE_RUNNING = 'Running'
def create_instance_action():
    instance_id = create_after_pay_instance(image_id=IMAGE_ID, instance_type=
INSTANCE_TYPE,
        security_group_id=SECURITY_GROUP_ID)
    check_instance_running(instance_id=instance_id)
def create_prepay_instance_action():
    instance_id = create_prepay_instance(image_id=IMAGE_ID, instance_type=INSTANCE_T
YPE,
        security_group_id=SECURITY_GROUP_ID)
    check_instance_running(instance_id=instance_id)
def create_after_pay_instance(image_id, instance_type, security_group_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_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;
def create_prepay_instance(image_id, instance_type, security_group_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_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_id, status=INSTANCE_RUNNING)
    index = 0
    while detail is None and index < 60:
        detail = get_instance_detail_by_id(instance_id=instance_id);
        time.sleep(10)
        if detail and detail.get('Status') == 'Stopped':
            logging.info("instance %s is stopped now.")
            start_instance(instance_id=instance_id)
            logging.info("start instance %s job submit.")
            detail = get_instance_detail_by_id(instance_id=instance_id, status=INSTANCE_RUNNING)
        while detail is None and index < 60:
            detail = get_instance_detail_by_id(instance_id=instance_id, status=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_InstancedIds(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()
3.3 Manage instances

In addition to using the ECS console for resource creation and daily management, you can also use API to manage and customize resources. API allows you to manage and configure ECS instances with greater flexibility. Alibaba Cloud encapsulates API in an SDK to integrate ECS instance management into existing systems. This topic describes how to manage ECS instances through API based on Python development. You can develop ECS instances easily even if you do not have Python development experience.

Get the AccessKey for a RAM user

An AccessKey (AccessKey ID and AccessKey Secret) is required when you want to use API to manage ECS instances. To keep your cloud service secure, you have to create a RAM user and generate an AccessKey for it, and authorize the RAM user to manage ECS resources only. Then, you can use the RAM user and its AccessKey to manage ECS resources by using API.

Follow these steps to get the AccessKey for a RAM user:

1. Create a RAM user and get the AccessKey.
2. Grant permissions to the RAM user directly. To manage ECS resources, you have to grant AliyunECSFullAccess to the RAM user.

Install the ECS Python SDK

Make sure that the Python runtime environment has been installed. This article uses Python 2.7+.

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

If you do not have the permission, switch to `sudo` to continue.

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

The SDK version is 2.1.2.

Hello Alibaba Cloud

Create the file hello_ecs_api.py. To use SDK, you have to use the AccessKey of the RAM user to instantiate an AcsClient object.
The AccessKey allows the RAM user to access Alibaba Cloud APIs and give you full access to the user. Keep them safe.

```python
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.DescribeRegionsRequest import DescribeRegionsRequest
clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secret', 'cn-beijing')

You can develop your first application after the AcsClient object is instantiated. Query the list of regions that your account supports. For more information, see query the list of available regions.

```python
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()
```

In the command line, run `python hello_ecs_api.py` to obtain a list of supported regions. The output is similar to the following.

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

**Query the list of ECS instances in the current region**

The process for querying the instance list is similar to the region list. You only need to replace the input parameter DescribeRegionsRequest with `DescribeInstancesRequest`. For a full list of query parameters, see query an instance list.

```python
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)
```

---

Issue: 20200417
logging.info("current region include instance %s", result)
def _print_instance_id(item):
    instance_id = item.get('InstanceId');
    return instance_id

The output is as follows.

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

For a full list of APIs, see ECS API overview. If you want to query a list of disks, replace DescribeInstancesRequest with DescribeDisksRequest.

Complete code

The following is the complete code of the operations described in this document.

```python
# 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 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. 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] %(levelname)s %(message)s',
                    datefmt='%a, %d %b %Y %H:%M:%S')
clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secret', 'cn-beijing')
# 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')
        assert response is not None
        assert instance_list is not None
        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
3.4 Release an instance

One important feature of ECS is on-demand resource creation. You can create custom resources elastically on demand during peak service hours, and then release those resources after service computing is completed. This document describes how to easily release ECS instances and achieve elasticity.

This topic covers the following APIs:

- **DeleteInstance**
- **ModifyInstanceAutoReleaseTime**
- **StopInstance**
- **Instance list query API**

After an ECS instance is released, the physical resources used by the instance are recycled, including disks and snapshots. The data of the instance is completely lost and can never be recovered. If you want to retain the data, we recommend that you create snapshots of disks before releasing the ECS instance. The snapshots can be directly used to create a new ECS instance.

To release an ECS instance, you must stop it first. If any application is affected after the ECS instance is stopped, restart the instance.

**Stop an ECS instance**

Use the StopInstance interface to stop an ECS instance, regardless of the billing method of the instance. The stop command is as follows. When the ForceStop parameter is set to true, the ECS instance is stopped directly but data is not necessarily written to a disk, similar to power failure. Therefore, if you want to release an instance, set ForceStop to true.

```python
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 when it is not in the Stopped status, an error occurs:

{"RequestId":"3C6DEAB4-7207-411F-9A31-6ADE54C268BE","Hostid":"ecs-cn-hangzhou.aliyuncs.com","Code":"IncorrectInstanceStatus","Message":"The current status of the resource does not support this operation."}

When the ECS instance is in the **Stopped** status, you can release it. The API has only two request parameters:

- InstanceId: Instance ID
- Force: If this parameter is set to “true”, the ECS instance is released forcibly even when it is not in the **Stopped** status. Use caution when setting this parameter. Release by mistake may affect your services.

The request to release an ECS instance is as follows.

```python
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)
```
The following response is returned when an ECS instance is released successfully:

```
{"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 set time is reached, Alibaba Cloud releases your ECS instance automatically. Use the `ModifyInstanceAutoReleaseTime` to set the automatic release time for an ECS instance.

---

**Note:**

The automatic release time follows the ISO8601 standard in UTC time. The format is `yyyy-MM-ddTHH:mm:ssZ`. If the seconds place is not 00, it is automatically set to start from the current minute. The automatic release time must be at least half an hour later than the current time, and must not be more than 3 years since the current time.

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

Then you can use the `DescribeInstances` to query the automatic release time.

```python
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]
```

---

Issue: 20200417
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 service changes, run the following command to set the automatic release time to null.

set_instance_auto_release_time('i-1111')

Complete example code

Note:
Proceed with caution when releasing ECS instances.

# 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 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[lineno:%(lineno)d] %(levelname)s %(message)s', 
    datefmt='%a, %d %b %Y %H:%M:%S')
clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secret', 'cn-beijing')

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-***'.
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):
    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):
    request = ModifyInstanceAutoReleaseTimeRequest()
    request.set_InstanceId(instance_id)
    if time_to_release is not None:
        request.set_AutoReleaseTime(time_to_release)
    _send_request(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("Release ecs instance by Aliyun OpenApi!")
    set_instance_auto_release_time('i-1111', '2017-01-28T06:00:00Z')
    # set_instance_auto_release_time('i-1111')
    # stop_instance('i-1111')
If you want to learn other API operations in ECS, see ECS API operation.

### 3.5 Renew an instance

Lifecycle is important to ECS instances of the Subscription billing method. In addition to the ECS console or the ECS purchase page, Alibaba Cloud provides you with APIs to view the resource expiration time and renew your instance.

This topic involves the following key functions:

- Query ECS instances by expiration time.
- Renew instances.
- Query the automatic renewal time of an ECS instance.
- Set the automatic renewal time of an ECS instance.

Lifecycle is important to ECS instances in the Subscription mode. If you fail to renew your ECS instance on time, the instance may be locked or even released, thus affecting your service continuity. You can use APIs to view the resource expiration time and renew your instance.

This article covers the following APIs:

- `DescribeInstances`
- `ModifyInstanceAutoRenewAttribute`

**Query the instances that will expire within the specified time range**

Use the `DescribeInstances` interface to query the instances that will expire within the specified time range by setting the filter parameters `ExpiredStartTime` and `ExpiredEndTime`. The time parameters follow the ISO8601 standard in UTC time, using the format `yyyy-MM-ddTHH:mmZ`. The system returns a list of instances that will expire within the specified time range. If you want to filter by security group, add the security group ID.

```python
INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING = '2017-01-22T00:00Z'
INSTANCE_EXPIRED_END_TIME_IN_UTC_STRING = '2017-01-28T00:00Z'
def describe_need_renew_instance(page_size=100, page_number=1, instance_id=None, check_need_renew=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_EXPIRED_END_TIME_IN_UTC_STRING)
    if instance_id is not None:
        request.set_InstanceIds(instance_id)
    response = client.do_action_with_exception(request)
    return response
```
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)

Renew ECS instances

Only ECS instances in the Subscription mode can be renewed. Pay-As-You-Go instances cannot be renewed. Renewals must be paid by account balance or credit. Fee deduction and order creation are in sync with API execution. Make sure that there is sufficient balance in your account.

```python
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 completed, you can query the resource expiration time of the instance based on `InstanceId`. Because the API is executed asynchronously, the expiration time is updated in 10 seconds.

Enable automatic ECS instance renewal

Alibaba Cloud provides the automatic renewal function for ECS instances in the Subscription mode to help you reduce the cost of expired resource maintenance. Fee deduction for automatic renewal starts at 08:00:00 nine days before the expiration date. If fee deduction fails on the first day, the deduction process repeats on the following days in sequence until fees are deducted successfully or resources are locked after the nine-day period. Make sure that you have sufficient balance or credit amount in your account.

- Query automatic renewal setting

You can use OpenAPI to query and set automatic renewal. The API supports only ECS instances in the Subscription mode. If you use the API on a Pay-As-You-Go instance, an error is returned. You can query the automatic renewal status of up to 100 ECS instances
of the Subscription billing method at a time. Use commas to separate multiple instance IDs.

The input parameter of DescribeInstanceAutoRenewAttribute is the instance ID.

InstanceId: You can query up to 100 ECS instances in the Subscription mode at a time. Use commas to separate multiple instance IDs.

```python
# check the instances is renew or not
def describe_auto_renew(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_auto_renew('i-1111,i-2222')
```

The following content is returned:

```json
{"InstanceRenewAttributes":{"InstanceRenewAttribute":[{"Duration":0,"InstanceId":"i-1111","AutoRenewEnabled":false},{"Duration":0,"InstanceId":"i-2222","AutoRenewEnabled":false}]},"RequestId":"71FBB7A5-C793-4A0D-B17E-D6B426EA746A"}
```

If automatic renewal is set, the returned attribute `AutoRenewEnabled` is true. If automatic renewal is not set, the attribute is false.

**Enable and cancel automatic renewal for ECS instances**

To enable automatic renewal for ECS instances, three input parameters are required:

- **InstanceId**: You can set automatic renewal for up to 100 ECS instances of the Subscription billing method at a time. Use commas to separate multiple instance IDs.
- **Duration**: Set to 1, 2, 3, 6, or 12, in unit of Month.
- **AutoRenew**: Set to true to enable automatic renewal. Set to false to disable automatic renewal.

```python
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)
```

**Issue:** 20200417
_send_request(request)

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

{"RequestId":"7DAC9984-AAB4-43EF-8FC7-7D74C57BE46D"}

You can perform a query after successful renewal. The system returns the renewal duration and the status of automatic renewal (true/false).

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

**Complete example code**

```python
# 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 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. DescribeInstanceAutoRenewAttributeRequest import
from aliyunsdkecs.request.v20140526. DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526. ModifyInstanceAutoRenewAttributeRequest import
from aliyunsdkecs.request.v20140526. RenewInstanceRequest import RenewInstanceRequest
logging.basicConfig(level=logging.INFO,
    format='%a,%d %B %Y %H:%M:%S

clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secret', 'cn-beijing')
# data format in UTC, only support passed the value for minute, seconds is not support.
INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING = '2017-01-22T00:00Z'
INSTANCE_EXPIRED_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,
        check_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_ids)
        logging.info("%s execute renew action ready", instance_ids)
    def describe_need_renew_instance(page_size=100, page_number=1, check_need_renew=True, security_group_id=None):
        request = DescribeInstancesRequest()
        request.set_PageSize(page_size)
        request.set_PageNumber(page_number)
        request.set_CheckNeedRenew(check_need_renew)
        if security_group_id is not None:
            request.set_SecurityGroupId(security_group_id)
        return clt.do_action_with_exception(request)
    response = describe_need_renew_instance()
    if response.get('TotalCount') > 0:
        instance_ids = ''
        for item in response_list:
            instance_id = item.get('InstanceId')
            instance_ids += instance_id + ','
        renew_instance(instance_id=instance_ids)
        logging.info("%s execute renew action ready", instance_ids)
```

Issue: 20200417
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)

# check the instances is renew or not
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') + ','
    if len(failed_instance_ids) > 0:
        logging.error("instance %s auto renew not match expect %s\n, 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)

# if using the instance id can be found means the instance is not renew successfully.
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

# Renew an instance for a month
def renew_instance(instance_id, period='1'):
    need_renew = check_instance_need_renew(instance_id)
    if need_renew:
        _renew_instance_action(instance_id=instance_id, period=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\n, 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 whether there is any instance that needs to be renewed within the specified
    # time range.
    describe_need_renew_instance()
    # Renew an instance by direct fee deduction
    renew_instance('i-1111')
    # Query the status of automatic renewal
    # describe_instance_auto_renew_setting('i-1111,i-2222')
    # Set automatic instance renewal
    # setting_instance_auto_renew('i-1111,i-2222')

If you want to learn other API operations in ECS, see ECS API operation.

3.6 Query available resources for configuration changes

This topic describes how to query information about available resources before you change
the configurations of an ECS instance.

Background information

Before you call an API to modify the configuration of an ECS instance
(such as ModifyInstanceSpec or ModifyPrepayInstanceSpec), you can call
DescribeResourcesModification to:

• Query the information of available resources you can use to upgrade an ECS instance.
• Query the information of available resources you can use to upgrade a system disk.

Limits

If you need to change the configurations of one or more ECS instances of different network
types and different instance generations at a time, you can configure MigrateAcrossZone
=true in DescribeResourcesModification. However, such changes will modify the private
IP addresses of ECS instances of the Classic network type. If you are using phased-out
instance types, you must exercise caution when you upgrade a non-I/O-optimized instance
to an I/O-optimized instance. Depending on the network type or instance type family,
changes to the configurations of one or more ECS instances at a time may cause the
following results.
### Item subject to change

<table>
<thead>
<tr>
<th>Change configurations of a non-I/O-optimized Generation I instance across instance types</th>
<th>Change configurations of instances of other generations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic network</td>
<td>VPC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Private IP address</th>
<th>Will change</th>
<th>Will not change</th>
<th>Will change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive name (Linux only)</td>
<td>Basic disk (cloud)</td>
<td>Changes to xvda or xvdb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ultra disk (cloud_efficiency)</td>
<td>Changes to vda or vdb</td>
<td></td>
</tr>
<tr>
<td>Software license code</td>
<td>Will change</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Prerequisites

You have obtained the AccessKey and the region ID of the related resource. For more information, see Regions and zones and Create an AccessKey.

### Install the Python SDK

**Note:**

Make sure that your Python SDK version is v4.6.3 or later. To get the latest ECS SDK, go to GitHub Repo Alibaba Cloud.

The following examples show how to install the Python SDK in the Linux (the version of Python is v2.7):

- For root users, run:

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

- For ordinary users, run:

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

**Note:**

Make sure that your Python SDK version is v4.6.3 or later. To get the latest ECS SDK, go to GitHub Repo Alibaba Cloud.
Scenario example

```python
# Your AccessKeyId
ak_id = "YOU_ACCESS_KEY_ID"

# Your AccessKeySeret
ak_secret = "YOU_ACCESS_SECRET"

# ID of the region to which the resource belongs
region_id = "cn-hangzhou"

# Resource ID, such as an instance ID
resource_id = "i-xxxx"

# Target resource type. Valid values: InstanceType| SystemDisk
destination_instance_type = 'InstanceType'
destination_system_disk = 'SystemDisk'

# Actions for changing configurations. Valid values: Upgrade | Downgrade | RenewDongrade | RenewModify
operation_type = 'Upgrade'

# Target instance type. You can call DescribeInstanceTypes to get the latest instance types. When the DestinationResource parameter takes the value of SystemDisk, the InstanceType parameter must be specified.
instance_type = "ecs.n1.large"

# Exercise caution when configuring the MigrateAcrossZone parameter. MigrateAcrossZone (Boolean) allows you to change configurations across one or more instances. Default value: False
# If the parameter MigrateAcrossZone takes the value of True, after you have upgraded your ECS instance according to the returned information, note the following:
# Classic network instances:
# 1. For a Generation I ECS instance, if it is upgraded from a non-I/O-optimized instance to an I/O-optimized instance, its private IP address, drive name, and software license code will change. For Linux instances, the basic disk (cloud) will be identified as xvda or xvdb, and the ultra disk (cloud_efficiency) and standard SSD (cloud_ssd) will be identified as vda or vdb.
# 2. For other generations of instances, the private IP address will change.
# VPC instances: For a Generation I ECS instance, if it is upgraded from a non-I/O-optimized instance to an I/O-optimized instance, its drive name and software license code will change. For Linux instances, the basic disk (cloud) will be identified as xvda or xvdb, and the ultra disk (cloud_efficiency) and standard SSD (cloud_ssd) will be identified as vda or vdb.
migrate_across_zone = False

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

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\n%(zoneld, values)

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

    # Send 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)
            return response_detail
述职 Exception as e:
        logging.error(e)
        return response_detail

Scenario 1: Query available resources for upgrading an instance type

    # Query available resources for upgrading an instance type
    def describe_resource_instance_type():
        request = build_request()
        request.set_DestinationResource(destination_instance_type)
        _execute_request(request)

Scenario 2: Query available resources for upgrading a system disk

    # Query available resources for upgrading a system disk
    def describe_resource_system_disk():
        request = build_request()
        request.set_DestinationResource(destination_system_disk)
        request.set_InstanceType(instance_type)
        _execute_request(request)

Complete code

    # coding=utf-8
    # if the python sdk is not install, run 'sudo pip install aliyun-python-sdk-ecs'
    # if the python sdk is install, run 'sudo pip install --upgrade aliyun-python-sdk-ecs'
    # make sure the sdk version is 4.6.3. To check your version, run 'pip show aliyun-python-sdk-ecs' 'pip show aliyun-python-sdk-ecs' to check
import json
class logging:
    def __init__(self, level=logging.INFO, format='%(asctime)s %(filename)s[%(funcName)s:%(lineno)d] %(levelname)s: %(message)s', datefmt='%a, %d %b %Y %H:%M:%S'):
        self.level = level
        self.format = format
        self.datefmt = datefmt

# Your AccessKeyId
ak_id = "YOU_ACCESS_KEY_ID"

# Your AccessKeySecret
ak_secret = "YOU_ACCESS_SECRET"

# ID of the region to which the resource belongs
region_id = "cn-hangzhou"

# Resource ID, such as an instance ID
resource_id = "i-xxxx"

# Target resource type. Valid values: InstanceType| SystemDisk
destination_instance_type = 'InstanceType'
destination_system_disk = 'SystemDisk'

# Actions for changing configurations. Valid values: Upgrade | Downgrade | RenewDowngrade | RenewModify
operation_type = 'Upgrade'

# Target instance type. You can call DescribeInstanceTypes to get the latest instance types. When the DestinationResource parameter takes the value of SystemDisk, the InstanceType parameter must be specified.
instance_type = "ecs.n1.large"

# Exercise caution when configuring the MigrateAcrossZone parameter. MigrateAcrossZone (Boolean) allows you to change configurations across one or more instances. Default value: False
migrate_across_zone = False

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

# Query available resources for upgrading an instance type
def describe_resource_instance_type():
    request = build_request()
    request.set_DestinationResource(destination_instance_type_type)
# Query available resources for upgrading a system disk
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')
            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":(zoneld, values)

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

# Send the 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"
    # describe_resource_instance_type()
    # describe_resource_system_disk()
    # describe_resource_instance_type()
# describe_resource_system_disk()