

阿里云 云服务器 ECS

SDK示例

文档版本：20190919

法律声明

阿里云提醒您 在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

| 格式 | 说明 | 样例 |
|---|-----------------------------------|--|
|  | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  禁止： 重置操作将丢失用户配置数据。 |
|  | 该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。 |
|  | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。 |  说明： 您也可以通过按Ctrl + A选中全部文件。 |
| > | 多级菜单递进。 | 设置 > 网络 > 设置网络类型 |
| 粗体 | 表示按键、菜单、页面名称等UI元素。 | 单击 确定 。 |
| <code>courier</code> 字体 | 命令。 | 执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。 |
| <code>##</code> | 表示参数、变量。 | <code>bae log list --instanceid</code> <code>Instance_ID</code> |
| <code>[]</code> 或者 <code>[a b]</code> | 表示可选项，至多选择一个。 | <code>ipconfig [-all -t]</code> |
| <code>{ }</code> 或者 <code>{a b}</code> | 表示必选项，至多选择一个。 | <code>swich {stand slave}</code> |

目录

| | |
|-------------------------|----|
| 法律声明..... | I |
| 通用约定..... | I |
| 1 SDK概述..... | 1 |
| 2 SDK使用方法和具体代码编写步骤..... | 2 |
| 3 Java示例..... | 4 |
| 3.1 安装Java SDK..... | 4 |
| 3.2 通用流程..... | 7 |
| 3.3 创建ECS实例..... | 11 |
| 3.3.1 批量创建ECS实例..... | 11 |
| 3.3.2 创建抢占式实例..... | 14 |
| 3.4 查询ECS实例..... | 19 |
| 3.5 变更ECS实例规格..... | 20 |
| 3.6 释放ECS实例..... | 22 |
| 4 Python示例..... | 24 |
| 4.1 批量创建ECS实例..... | 24 |
| 4.2 弹性创建ECS实例..... | 28 |
| 4.3 弹性管理ECS实例..... | 34 |
| 4.4 弹性释放ECS实例..... | 37 |
| 4.5 ECS实例续费..... | 42 |
| 4.6 管理抢占式实例..... | 47 |
| 4.7 查询可用的变配资源..... | 54 |

1 SDK概述

本文介绍了云服务器ECS提供的SDK语言版本，列举了最新版本SDK的获取地址。

SDK简介

云服务器ECS SDK封装了2014-05-06版本API，以访问密钥（AccessKey）识别调用者身份，提供自动签名等功能，方便您通过API创建和管理资源。

实现云服务器ECS功能需要您同时安装ECS SDK和阿里云核心库。以Java SDK为例，您需要下载aliyun-java-sdk-core以及aliyun-java-sdk-ecs。

SDK列表

下表提供了云服务器ECS支持的SDK列表，您可以在GitHub仓库查看SDK更新历史、获取安装包以及查看指导文档。

| 编程语言 | GitHub地址 | GitHub文档 |
|---------|---|---------------------------|
| Java | aliyun-openapi-java-sdk | README.md |
| Python | aliyun-openapi-python-sdk | README.md |
| PHP | aliyun-openapi-php-sdk | README.md |
| .NET | aliyun-openapi-net-sdk | README.md |
| C++ | aliyun-openapi-cpp-sdk | README.md |
| Go | alibaba-cloud-sdk-go | README.md |
| Node.js | aliyun-openapi-nodejs-sdk | README.md |

相关链接

- 阿里云开发者中心汇聚了高频使用的编程语言SDK，提供了Maven项目地址、pip命令、版本说明等信息。更多详情，请访问[开发者中心](#)。
- 更多语言版本的SDK，请前往[第三方SDK服务](#)中选择。
- 如何获取AccessKey请参见[#unique_4](#)。
- 云服务器ECS的服务接入地址（Endpoint）列表请参见[#unique_5](#)。

2 SDK使用方法和具体代码编写步骤

新版 SDK 的文件名通常以 aliyun-XXXX-sdk 开头，后面跟上产品名称如 ECS，组成如 aliyun-java-sdk-ecs 的包名。其中有一个核心包 aliyun-java-sdk-core，其中封装了所有产品的 SDK 都会用到的一些类，如 IClientProfile 类、IAcsClient 类、异常类等。产品相关的类均以产品为单位打包成不同名称的 Jar 包。

您需要准备好您的 AccessKey，用于输出到 [创建 Profile](#) 中。

Java SDK 使用方法示例

以 ECS Java SDK 查询可用镜像资源的方法 [#unique_8](#) 为例，介绍 SDK 使用的完整流程，其中 IClientProfile 和 IAcsClient 两个类包含在 aliyun-java-sdk-core 包中，其他的类均包含在 aliyun-java-sdk-ecs 包中。

1. 创建 Profile。生成 IClientProfile 的对象 profile，该对象存放 AccessKeyId 和 AccessKeySecret 和默认的地域信息，如示例中的 cn-hangzhou，更多关于地域的信息，参阅 [#unique_9](#)。

```
IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", ak, aks); #ak 是您的 AccessKey, aks 是您的 AccessKeySecret
```

2. 创建 Client。从 IClientProfile 类中再生成 IAcsClient 的对象 client，后续获得 response 都需要从 IClientProfile 中获得。

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

3. 创建 Request。创建一个对应方法的 Request，类的命名规则一般为 API 的方法名加上 Request，如获得镜像列表的 API 方法名为 [#unique_8](#)，那么对应的请求类名就是 DescribeImagesRequest，直接使用构造函数生成一个默认的对象 describe。

```
DescribeImagesRequest describe = new DescribeImagesRequest();
```

4. 设置 Request 的参数。请求类生成好之后需要通过 Request 类的 setXxx 方法设置必要的信息，即 API 参数中必须要提供的信息，[#unique_8](#) 的 API 方法必须要提供的参数为 RegionId，该值可以省略，因为 IClientProfile 中已经提供了地域信息，同样的也可

以通过 `setXxx` 方法设置其他可选的参数，如这里设置要查询的镜像为自定义镜像，则设置 `ImageOwnerAlias` 的值为 `self`，表示查询您的自定义镜像。

```
describe.setImageOwnerAlias("self");
```

5. 参数设置完毕后，通过 `IACSClient` 对象获得对应 `Request` 的响应。

```
DescribeImagesResponse response = client.getAcsResponse(describe);
```

6. 在 `Response` 中获得返回的参数值。接着可以调用 `response` 中对应的 `getXxx` 方法获得返回的参数值了，如获得某个镜像的名字。根据 API 方法的不同，返回的信息中可能会包含多层的信息，如获得镜像列表这个方法，返回的信息中镜像是以一个集合来表示的，集合中存放了每个镜像的信息，对于 Java SDK 而言，那么存放镜像信息的就是一个列表，需要先通过 `getImages()` 获得 `Image` 对象的集合，然后再通过遍历等方法取得其中某个镜像的信息，之后调用 `getXxx` 方法获得具体的信息。

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

至此，一个完整的调用就完成了。

PHP SDK 注意事项

使用 PHP SDK 和 Java SDK 的类似，可以归纳为：

1. 创建 Profile。
2. 创建 Client。
3. 创建 Request。
4. 设置 Request 的参数。
5. 使用 Client 对应的方法传入 Request，获得 Response。
6. 在 Response 中获得返回的参数值。

Python SDK 注意事项

使用 Python SDK 省略了创建 Profile 这一步，直接创建 Client，然后执行后面的步骤即可。

参考信息

- 关于 ECS 的所有 API，请参阅 [#unique_10](#)。
- 关于如何创建 AccessKey，请参阅 [创建 AccessKey](#)。

3 Java示例

3.1 安装Java SDK

本文介绍了如何安装ECS Java SDK以及阿里云核心库。

前提条件

安装ECS Java SDK必须用1.6或更高版本JDK。

背景信息

您可以自行下载aliyun-java-sdk-ecs，也可以使用Maven坐标来使用SDK。

本文示例中，客户端使用了Windows 7 64位操作系统，安装的Java使用环境为Eclipse Luna。

操作步骤

1. 访问Maven官方下载页面（[Download Apache Maven](#)）下载对应操作系统的Maven软件。

Checksum文件可供您校验下载文件是否正确无误。

| Files | Link | Checksums | Signature |
|-----------------------|---|--|---|
| Binary tar.gz archive | apache-maven-3.6.1-bin.tar.gz | apache-maven-3.6.1-bin.tar.gz.sha512 | apache-maven-3.6.1-bin.tar.gz.asc |
| Binary zip archive | apache-maven-3.6.1-bin.zip | apache-maven-3.6.1-bin.zip.sha512 | apache-maven-3.6.1-bin.zip.asc |
| Source tar.gz archive | apache-maven-3.6.1-src.tar.gz | apache-maven-3.6.1-src.tar.gz.sha512 | apache-maven-3.6.1-src.tar.gz.asc |
| Source zip archive | apache-maven-3.6.1-src.zip | apache-maven-3.6.1-src.zip.sha512 | apache-maven-3.6.1-src.zip.asc |

2. 找到并解压Maven压缩包，将下列Maven库信息添加到conf文件夹下的`settings.xml`文件中。

```
<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>
```

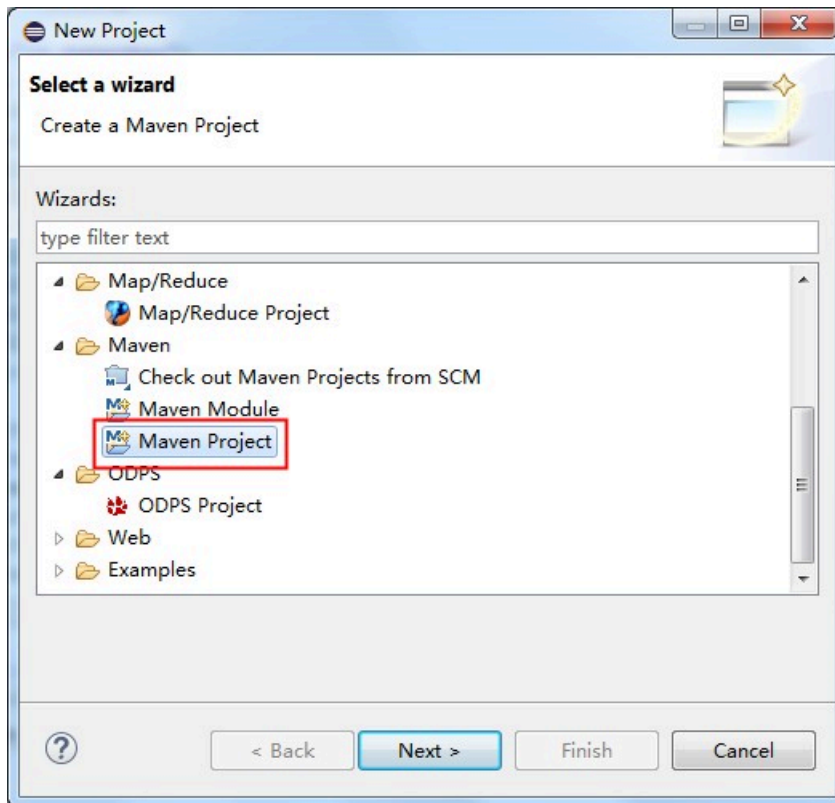


```
</repositories>
```

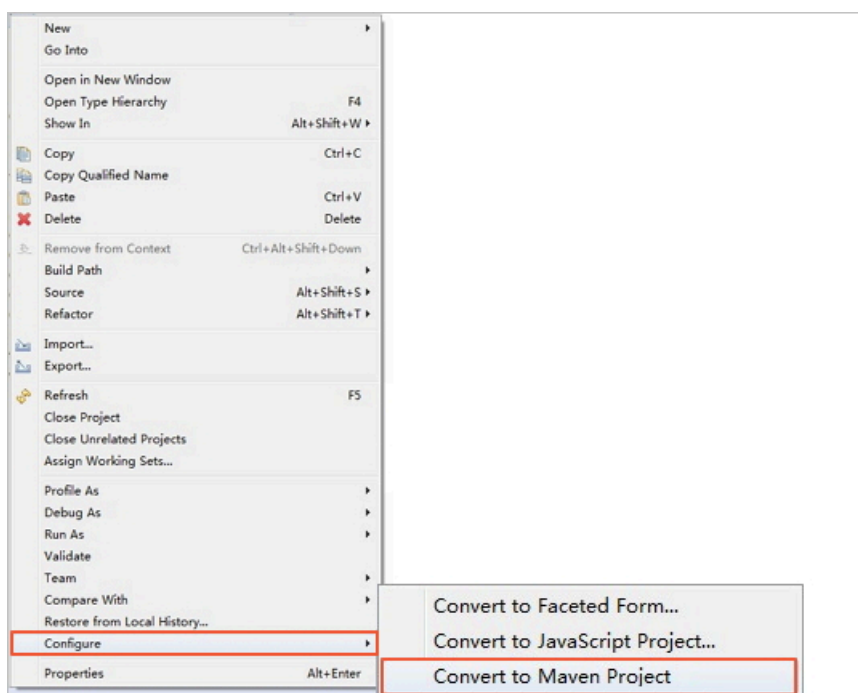
3. 打开阿里云开发者中心 (<https://developer.aliyun.com/tools/sdk?#/java>)，将阿里云的SDK存放的Maven库加入到Maven软件中。

4. 通过以下任一方式创建Maven项目：

- 方式一：在Eclipse中添加一个Maven项目。



- 方式二：将已有的项目转换为Maven项目。

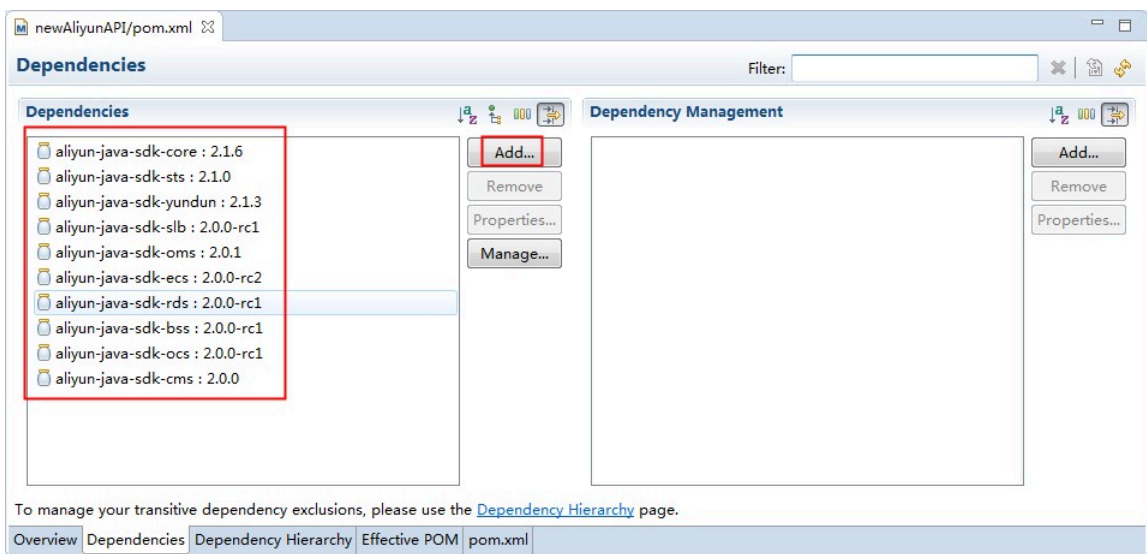


5. 通过以下任一方式将dependency加入Maven项目中:

- 方式一: 打开Maven项目下的pom.xml文件, 添加aliyun-java-sdk-core、aliyun-java-sdk-ecs和fastjson依赖。您可以根据自身需要添加不同的依赖。

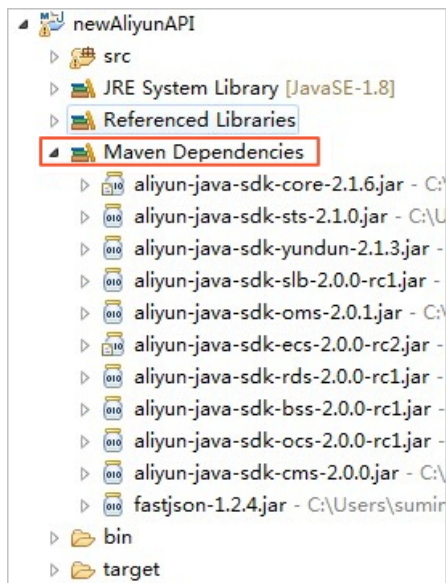
```
<dependencies>  
  <dependency>  
    <groupId>com.aliyun</groupId>  
    <artifactId>aliyun-java-sdk-core</artifactId>  
    <version>4.4.3</version>  
  </dependency>  
  <dependency>  
    <groupId>com.aliyun</groupId>  
    <artifactId>aliyun-java-sdk-ecs</artifactId>  
    <version>4.17.1</version>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

- 方式二: 通过图形化界面选择性添加。



6. 保存更改。

Maven Dependencies中会自动下载并加入.jar后缀的相应Java SDK。



3.2 通用流程

以云服务器ECS Java SDK查询可用镜像资源的方法DescribeImages为例，介绍编写ECS Java SDK的通用流程。

前提条件

您已经创建了AccessKey。详细步骤请参见[#unique_4](#)。



说明:

为避免主账号泄露AccessKey带来的安全风险，建议您创建RAM用户，授予RAM用户云服务器ECS相关的访问权限，再使用RAM用户的AccessKey调用SDK。详情请参见[#unique_14](#)。

背景信息

- 本文示例中，IClientProfile和IAcsClient两个类包含在aliyun-java-sdk-core中，其他的类均包含在aliyun-java-sdk-ecs中。
- 本文示例目的是查询云服务器ECS公共镜像，您可以参见镜像相关文档了解什么是公共镜像。更多详情，请参见[#unique_15](#)。
- 下表列举了旧版SDK与新版SDK的方法、类以及对象之间的区别。若您使用的是旧版SDK，建议您切换为新版SDK，获取最新功能。

| 对比项 | 新版SDK | 旧版SDK |
|------|------------------|-----------|
| 提交请求 | getAcsResponse() | execute() |

| 对比项 | 新版SDK | 旧版SDK |
|---------------|--|--|
| 存放AccessKey的类 | IClientProfile | AliyunClient |
| 存放身份凭据对象 | DefaultProfile.getProfile (RegionId, AccessKey, AccessKeySecret) | new DefaultAliyunClient (APIUrl, AccessKey, AccessKeySecret) |
| 包名前缀 | com.aliyuncs | com.aliyun.api |

操作步骤

1. 从IClientProfile类中生成对象Profile。

Profile对象存放地域、AccessKeyID和AccessKeySecret，如示例中的cn-hangzhou。更多关于地域的信息，请参见[#unique_9](#)。

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

2. 从IClientProfile类中生成IAcsClient的对象client。

后续获得response从IClientProfile中获得。

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

3. 创建一个对应方法的Request，使用构造函数生成一个默认的类型request。

类的命名规则为API方法名加上Request。获得镜像列表的API方法名为DescribeImages，对应的请求类名就是DescribeImagesRequest。

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

4. 设置请求类request的参数。

通过request类的setXxx方法设置必要的信息，即API中必须要提供的信息，通过setXxx方法设置参数。示例中：

- DescribeImages的API方法RegionId用来指定地域。
- DescribeImages的API方法ImageOwnerAlias用来设置要查询的镜像类型。设置setImageOwnerAlias的值为system，表示查询公共镜像。

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

5. 通过client对象获得对应request响应。

```
DescribeImagesResponse response = client.getAcsResponse(request);
```

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

6. 调用response中对应的getXxx方法获得返回的参数值。

假设您需要获取某份镜像的名字，先通过getImages()获得Image对象的集合，再通过遍历等方法取得其中某个镜像的信息，调用getImageName()或者getImageId获得具体的信息。

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

根据API方法的不同，返回的信息中可能会包含多层的信息，如DescribeImages返回的信息中镜像是以一个集合来表示的，集合中存放了每个镜像的信息，汇聚成一个列表。您需要通过getImages()获得Image对象的集合，通过遍历等方法取得其中某个镜像的信息，再调用getXxx方法获得具体的信息。

7. 使用catch()处理服务器报错和客户端报错。

- 服务端报错

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

- 客户端报错

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

预期结果

- 全部返回信息如下所示：

```
{
  "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位",
```

```

        "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
    }
}
]
}
}

```

- 获取具体返回参数的查询结果，例如ImageId和ImageName：

```

freebsd_11_02_64_30G_alibase_20190722.vhd
freebsd_11_02_64_30G_alibase_20190722.vhd

```

完整代码示例

以下为本文示例的完整Java SDK代码。

```

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*****", "BNP01zoNSi484oizGM9fzzwJJ
*****");
        IAcsClient client = new DefaultAcsClient(profile);

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

        try {

```

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

相关文档

[#unique_16](#)

3.3 创建ECS实例

3.3.1 批量创建ECS实例

本文介绍如何通过阿里云ECS Java SDK调用RunInstances创建一台或多台ECS实例。

前提条件

创建ECS实例前，您必须提前查询以下信息：

- 调用DescribeRegions查询ECS实例启动的阿里云地域，假设为cn-hangzhou。
- 调用DescribeImages接口查看要使用的镜像ID，假设为freebsd_11_02_64_30G_alibase_20190722.vhd。
- 调用DescribeInstanceTypes查询ECS实例使用的实例规格，假设为ecs.g5.large。详情请参见[#unique_19](#)。
- 调用DescribeSecurityGroups查询指定地域中的一个或多个安全组ID，假设为sg-bp1fg655nh68xyz9i***。安全组的网络类型将决定ECS实例的网络类型，例如，如果您选择的是一个专有网络VPC类型的安全组，则新建的ECS实例会自动加入安全组所属的专有网络VPC。
- 如果安全组的网络类型为专有网络VPC，调用VPC API DescribeVSwitches查询安全组所属专有网络VPC中的虚拟交换机ID，假设为vsw-bp1wt4qpuavdb6y6k8***。
- 使用按量付费的ECS资源时，您的阿里云账户余额不得少于100.00元人民币或者等值的代金券或者优惠券。

背景信息

本文调用RunInstances接口创建并自动启动多台ECS实例。详情请参见[#unique_20](#)。



说明:

调用RunInstances会创建ECS实例等计费资源，会产生实际费用。如果您只需测试示例代码，可以在代码中设置DryRun方法，只发送检查请求，不会创建实例。检查项包括是否填写了必需参数、请求格式、业务限制和ECS库存等。

代码示例

以下代码适用于公网带宽采用按流量计费、实例计费方式采用按量付费、网络类型采用专有网络VPC的新建ECS实例：

```
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) {
        // 创建并初始化DefaultAcsClient实例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-
hangzhou", "<yourAccessKeyId>", "<yourAccessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        // 创建API请求，并设置参数。
        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());

        // 添加一块数据盘，数据盘类型为SSD云盘，容量为100GiB，并开启了随ECS实例
释放。
        List<RunInstancesRequest.DataDisk> dataDiskList = new
ArrayList<RunInstancesRequest.DataDisk>();

        RunInstancesRequest.DataDisk dataDisk1 = new RunInstanc
esRequest.DataDisk();
        dataDisk1.setSize(100);
        dataDisk1.setCategory("cloud_ssd");
        dataDisk1.setDeleteWithInstance(true);
        dataDiskList.add(dataDisk1);
```



```

request.setDataDisks(dataDiskList);
// 批量创建五台ECS实例，如果不设置该参数，默认创建一台ECS实例。
// request.setAmount(5);
// 如果缺少库存可以接受的最低创建数量。
// 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);
// 打开预检参数功能，不会实际创建ECS实例，只检查参数正确性、用户权限或者
ECS库存等问题。
// 实际情况下，设置了DryRun参数后，Amount必须为1，MinAmount必须为
空，您可以根据实际需求修改代码。
request.setDryRun(true);
request.setInstanceChargeType("PostPaid");

// 发起请求并处理返回或异常。
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());
}
}
}

```

预期结果

实际返回结果为：

```

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

```

相关文档

[#unique_20](#)
[#unique_21](#)
[#unique_16](#)
[#unique_22](#)
[#unique_23](#)

#unique_24

3.3.2 创建抢占式实例

本文介绍了如何通过阿里云ECS Java SDK调用DescribeAvailableResource和DescribeSpotPriceHistory查询抢占式实例的供应情况和历史低价，再调用CreateInstance创建抢占式ECS实例。

前提条件

- 您使用的aliyun-java-sdk-ecs版本必须大于等于4.2.0。
- 创建抢占式ECS实例的镜像必须包含了实例运行的所以环境依赖。建议您使用自定义镜像创建实例，保证抢占式ECS实例能快速处理业务数据。假设为m-bp146shijn7hujkui9***。
- 调用DescribeRegions查询ECS实例启动的阿里云地域，假设为cn-hangzhou。
- 调用DescribeInstanceTypes查询ECS实例使用的实例规格，假设为ecs.g5.large。详情请参见[#unique_19](#)。

背景信息

抢占式实例能够帮助您建立低价处理业务数据的方案。创建抢占式实例时，您需要指定出价模式，当指定的实例规格当前市场价格低于您的出价时，您就能成功创建抢占式实例，并按当前市场价格计费。抢占式实例常被运用于控制计算成本、适用于无状态的应用场景。更多详情，请参见[#unique_26](#)。

操作步骤

1. 封装一个ApiCaller.java类，完成初始化profile和client，以及错误处理等功能。

```
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. 调用DescribeAvailableResource查询指定地域中的抢占式实例提供的实例规格。

```
public class DescribeAvailableResourceSample {  
    public static void main(String[] args) {  
        ApiCaller caller = new ApiCaller();  
        DescribeAvailableResourceRequest request = new DescribeAv  
ailableResourceRequest();  
        // 通过DescribeRegionsRequest获取地域ID。  
        request.setRegionId("cn-hangzhou");  
        request.setDestinationResource("InstanceType");  
        // 设置为按量付费, 必填。  
        request.setInstanceChargeType("PostPaid");  
        // 查询是否可以购买抢占式实例, 必填。  
        request.setSpotStrategy("SpotAsPriceGo");  
        // 不设置可用区ID则默认查询所有可以创建抢占式实例的可用区。  
        request.setZoneId("cn-hangzhou-h");  
        // 不设置实例规格则默认查询所有可以创建抢占式实例的实例规格。  
        request.setInstanceType("ecs.g5.large");  
        request.setSystemDiskCategory("cloud_ssd");  
        request.setNetworkCategory("vpc");  
  
        DescribeAvailableResourceResponse response = caller.doAction  
(request);  
        System.out.println(JSON.toJSONString(response));  
    }  
}
```

输出结果示例:

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

```
}
```

3. (可选) 调用DescribeSpotPriceHistory查询抢占式实例历史价格变化。



说明:

DescribeSpotPriceHistory最多允许获取30天内的价格变化。

```
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");// 地域必填。
            request.setZoneId("cn-hangzhou-b");// 可用区必填。
            request.setInstanceType("ecs.g5.large");// 实例规格必填。
            request.setNetworkType("vpc");// 网络类型必填。
            // request.setStartTime("2017-09-20T08:45:08Z");// 价格开始时间, 选填, 默认3天内数据。
            // request.setEndTime("2017-09-28T08:45:08Z");// 价格结束时间, 选填。
            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 {
        }
    }
}
```

输出结果示例:

```
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. 调用CreateInstances创建一台抢占式实例。

如果您需要批量创建多台抢占式实例，可以调用RunInstance。详情请参见[批量创建ECS实例](#)。

```
public class CreateInstancesSample {
    public static void main(String[] args) {
        ApiCaller caller = new ApiCaller();
        CreateInstancesRequest request = new CreateInstancesRequest
        ();
        request.setRegionId("cn-hangzhou");// 地域ID。
        request.setZoneId("cn-hangzhou-h");// 可用区ID。
        request.setSecurityGroupId("sg-bp11nhf94ivkdxwb2***");// 安全
        组ID。
        request.setImageId("m-bp146shijn7hujkui9***");// 建议您使用自
        定义镜像，可以快速承载业务。
        request.setVSwitchId("vsw-bp164cyonthfudn9kj***");// 与安全组
        同属于一个专有网络VPC的虚拟交换机ID。

        request.setInstanceType("ecs.g5.large");// 填入您询价后需要购
        买的规格。
        request.setSystemDiskCategory("cloud_ssd");// 系统盘类型，支持
        cloud_essd, cloud_ssd, cloud_efficiency, cloud。
        request.setSystemDiskSize(40);

        request.setInstanceChargeType("PostPaid");// 设置为按量付费，必
        填。
        request.setSpotStrategy("SpotWithPriceLimit");// SpotWithPr
        iceLimit出价模式，SpotAsPriceGo不用出价，最高按量付费价格。
        request.setSpotPriceLimit(0.25F);// SpotWithPriceLimit出价模
        式生效，您能接受的最高价格，必须高于市场成交价才能运行实例。
        // request.setAmount(2);// 创建两台抢占式实例，不设置该参数则默认创
        建一台实例。

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

5. 回收抢占式实例。



说明:

由于抢占式实例的生命周期较短，您需要在成功创建抢占式实例的一小时内完成处理中断的工作，避免抢占式实例被释放带来的不便。

- a) 在ECS实例操作系统内运行以下命令，通过实例元数据查询抢占式实例的释放时间。

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



说明:

如果返回信息为空，说明抢占式实例可以继续被使用。如果返回类似2015-01-05T18:02:00Z格式的信息（UTC+0时间），说明抢占式实例将于该时间点被回收。

- b) 调用DescribeInstances根据返回的OperationLocks判断实例是否进入待回收状态。

```
public class DescribeInstancesSample {
    public static void main(String[] args) throws InterruptedException {
        ApiCaller caller = new ApiCaller();
        JSONArray allInstances = new JSONArray();
        allInstances.addAll(Arrays.asList("i-bp18hgfai8ekoqwo0***",
            "i-bp1ecbyds24ij63w1***"));
        while (!allInstances.isEmpty()) {
            DescribeInstancesRequest request = new DescribeInstancesRequest();
            request.setRegionId("cn-hangzhou");
            request.setInstanceIds(allInstances.toJSONString()); // 指定实例ID，提高效率。
            DescribeInstancesResponse response = caller.doAction(request);
            List<DescribeInstancesResponse.Instance> instanceList = response.getInstances();
            if (instanceList != null && !instanceList.isEmpty()) {
                for (DescribeInstancesResponse.Instance instance : instanceList) {
                    System.out.println("result:instance:" + instance.getInstanceId() + ",was created in zone:" + instance.getZoneId());
                    if (instance.getOperationLocks() != null) {
                        for (DescribeInstancesResponse.Instance.LockReason lockReason : instance.getOperationLocks()) {
                            System.out.println("instance: " + instance.getInstanceId() + "-->lockReason:" + lockReason.getLockReason() + ",vmStatus:" + instance.getStatus());
                            if ("Recycling".equals(lockReason.getLockReason())) {
                                // 输出回收信息并从实例ID列表中删除被回收的实例ID。
                                System.out.println("Preemptible instance will be recycled immediately, instance id: " + instance.getInstanceId());
                                allInstances.remove(instance.getInstanceId());
                            }
                        }
                    }
                }
            }
            System.out.print("Try describeInstancesRequest again later...");
        }
    }
}
```

```
        Thread.sleep(2 * 60 * 1000);
    } else {
        break;
    }
}
}
```

触发回收抢占式实例输出结果如下：

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

相关文档

[#unique_21](#)
[#unique_27](#)
[#unique_28](#)
[#unique_22](#)
[#unique_29](#)
[#unique_30](#)
[#unique_20](#)

3.4 查询ECS实例

本文介绍了如何通过阿里云ECS Java SDK调用DescribeInstances根据条件筛选ECS实例。

前提条件

您必须至少创建了一台ECS实例。详细步骤请参见[批量创建ECS实例](#)。

背景信息

查询ECS实例适用于在众多实例中快速筛选出需要操作的实例的所需信息，例如：

- 在根据实例ID修改公网带宽前，根据实例计费方式、运行状态、公网带宽计费方式等筛选出符合条件的ECS实例。
- 在更新实例上部署的应用前，查询所有使用了相同镜像的ECS实例。

代码示例

以下代码适用于查询中国杭州地域下公网带宽采用按流量计费、实例计费方式采用按量付费、网络类型采用专有网络VPC的ECS实例：

```
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) {
        // 创建DefaultAcsClient实例并初始化。
        DefaultProfile profile = DefaultProfile.getProfile("cn-
hangzhou", "<yourAccessKeyId>", "<yourAccessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        // 创建API请求并设置参数。
        DescribeInstancesRequest request = new DescribeInstancesReq
uest();
        request.setRegionId("cn-hangzhou");
        request.setInstanceNetworkType("vpc");
        request.setInstanceChargeType("PostPaid");
        request.setInternetChargeType("PayByTraffic");
        request.setPageSize(10);

        try {
            // 发起请求并处理应答或异常。
            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());
        }
    }
}
```

预期结果

由于以上代码指定获取镜像ID、实例ID以及公网IP地址，则实际返回结果为：

```
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
```

相关文档

[#unique_27](#)

3.5 变更ECS实例规格

本文通过阿里云ECS Java SDK调用ModifyInstanceSpec变更一台按量付费ECS实例的实例规格，调用ModifyPrepayInstanceSpec变更一台包年包月ECS实例的实例规格。

前提条件

变更ECS实例规格时，您需要注意：

- 按量付费实例不能处于欠费状态。
- 包年包月实例不能处于过期状态。
- 实例必须处于已停止（Stopped）状态。

变更包年包月实例规格

以下代码适用于在中国杭州地域下，通过调用ModifyPrepayInstanceSpec将一台包年包月ECS实例的实例规格变更为ecs.g5.large：

```
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) {
        // 创建并初始化DefaultAcsClient实例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-
hangzhou", "<yourAccessKeyId>", "<yourAccessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建API请求，并设置参数。
        ModifyPrepayInstanceSpecRequest request = new ModifyPrep
ayInstanceSpecRequest();
        request.setRegionId("cn-hangzhou");
        request.setInstanceId("i-bp1jd3uddaduyo8*****");
        // 设置新的实例规格，ModifyPrepayInstanceSpecRequest支持升级实例规格
和降配实例规格。
        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);
    }
}
```

```
}
```

变更按量付费实例规格

以下代码适用于在中国杭州地域下，通过调用ModifyInstanceSpec将一台按量付费ECS实例的实例规格变更为ecs.g5.large：

```
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) {
        // 创建并初始化DefaultAcsClient实例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-
hangzhou", "<yourAccessKeyId>", "<yourAccessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建API请求，并设置参数。
        ModifyInstanceSpecRequest request = new ModifyInstanceSpecRe
quest();
        request.setRegionId("default");
        // 必须指定实例ID。
        request.setInstanceId("i-bp1gc5z6103qs2t40***");
        // 设置新的实例规格，ModifyInstanceSpecRequest支持升级实例规格和降配
实例规格。
        request.setInstanceType("ecs.g5.large");

        try {
            ModifyInstanceSpecResponse response = client.getAcsResp
onse(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());
        }
    }
}
```

相关文档

[#unique_33](#)

[#unique_34](#)

3.6 释放ECS实例

本文介绍了如何通过阿里云ECS Java SDK调用DeleteInstance删除一台ECS实例。

前提条件

您必须至少创建了一台ECS实例。详细步骤请参见[批量创建ECS实例](#)。

代码示例

以下代码适用于删除华东1（杭州）地域下的一台ECS实例：

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

public class DeleteInstance {

    public static void main(String[] args) {
        // 创建DefaultAcsClient实例并初始化
        DefaultProfile profile = DefaultProfile.getProfile("cn-
hangzhou", "<yourAccessKeyId>", "<yourAccessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        // 创建API请求并设置参数。
        DeleteInstanceRequest request = new DeleteInstanceRequest();
        // 设置一个地域ID。
        request.setRegionId("cn-hangzhou");
        // 指定一个实例ID。
        request.setInstanceId("i-bp1gvi17n5p8hav0i***");

        try {
            // 发起请求并处理应答或异常。
            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());
        }
    }
}
```

预期结果

实际返回结果为：

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

相关文档

[#unique_36](#)

[#unique_37](#)

4 Python示例

4.1 批量创建ECS实例

RunInstances批量创建实例接口可以帮助您一次创建多台ECS按量付费或包年包月实例来完成应用的开发和部署，方便实现弹性的资源创建。

前提条件

调用API前，您需要先创建AccessKey。具体操作，请参见[#unique_4](#)。



注意：

禁止使用主账号AK，因为主账号AK泄露会威胁您所有资源的安全。请使用子账号AK进行操作，可有效降低AK泄露的风险。

背景信息

和CreateInstance接口相比，RunInstances接口有以下优点：

- 单次可以最多创建100台实例，避免重复调用。
- 实例创建之后，实例会自动变成Starting状态，然后变成Running状态，不需要您调用StartInstance的操作。
- 创建实例的时候指定了InternetMaxBandwidthOut，则自动为您分配公网IP，不需要您再调用分配IP的操作。
- 您也可以一次创建100台抢占式实例，充分满足您的弹性需求。
- 创建的参数保持和CreateInstance保持兼容，增加了Amount参数来设定创建的个数，以及AutoReleaseTime参数来设定自动释放时间，不需要您再额外设置自动释放时间。
- 创建返回一个InstanceIdSets会记录相关的InstanceIds，您只需要根据实例ID轮询实例状态即可。详情请参见[#unique_40](#)。

安装ECS Python SDK

首先确保您已经具备Python的Runtime，本文中使用的Python版本为2.7+。

这里以Python为示例，其他SDK的版本大于4.4.3即可。

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

如果提示您没有权限，请切换sudo继续执行。

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

本文使用的SDK版本为4.4.3，如果您使用的是旧版本的SDK，需要更新。

批量创建实例

首先创建RunInstancesRequest的实例，然后填入相关需要的参数即可。

下面的例子创建了2台实例，并且添加了自动每隔10秒钟检查一次实例的运行状态。直到实例状态变成Running结束创建流程。

```
# 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"
# 选择的vswitchId
vswitch_id = "vws-xxxxx"
# 使用的镜像信息
image_id = "centos_7_03_64_20G_alibase_20170818.vhd"
# 当前vpc类型的安全组
security_group_id = "sg-xxxxx"
# instance number to launch, support 1-100, default value is 100
amount = 2;
# instance auto delete time 按照 ISO8601 标准表示，并需要使用 UTC 时间。格式
# 为 yyyy-MM-ddTHH:mm:ssZ 。最短在当前时间之后半小时。最长不能超过当前时间起三年
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('InstanceId
Set')
        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)
```

批量创建实例并自动分配公网IP

相比批量创建实例的代码，只需要添加一行属性，指定公网的带宽即可。下面的例子中默认给实例都分配了1 M的按流量带宽。

```
# 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)
```

批量创建实例并自动设置自动释放时间

相比批量创建实例的代码，只需要添加一行属性，指定实例的自动释放时间即可。自动释放时间按照ISO8601标准表示，并需要使用UTC时间，格式为yyyy-MM-ddTHH:mm:ssZ。最短在当前时间之后半小时，最长不能超过当前时间起三年。详情请参见[#unique_41](#)。

```
# 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)
```

完整代码示例

以上操作完整的代码示例如下所示。

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

```
# make sure the sdk version is 4.4.3, you can use command '
pip show aliyun-python-sdk-ecs' to check
import json
import logging
import time
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest
    import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.RunInstancesRequest
import RunInstancesRequest
logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s %(filename)s[line:%(
lineno)d] %(levelname)s %(message)s',
                    datefmt='%a, %d %b %Y %H:%M:%S')

# your 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"
# 选择的vswitchId
vswitch_id = "vws-xxxxx"
# 使用的镜像信息
image_id = "centos_7_03_64_20G_alibase_20170818.vhd"
# 当前vpc类型的安全组
security_group_id = "sg-xxxxx"
# instance number to launch, support 1-100, default value is
100
amount = 2;
# instance auto delete time 按照 ISO8601 标准表示, 并需要使用
UTC 时间. 格式为 yyyy-MM-ddTHH:mm:ssZ 。 最短在当前时间之后半小时。
最长不能超过当前时间起三年
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()
```

4.2 弹性创建ECS实例

本文以Python SDK为例，为您演示如何通过CreateInstance等阿里云API弹性地创建和管理ECS实例。

API列表

本文创建ECS实例时使用的API如下所示：

- [CreateInstance](#)：创建ECS实例
- [DescribeInstances](#)：查询实例列表
- [StartInstance](#)：启动ECS实例
- [StopInstance](#)：停止ECS实例
- [AllocatePublicIpAddress](#)：分配公网IP地址

创建按量付费ECS实例

前提条件

使用按量付费服务，您的账户账号不得少于100元，您可以在[阿里云费用中心](#)完成账号充值。

必选属性

- SecurityGroupId: 安全组ID。安全组相当于虚拟防火墙，通过安全组规则控制和保护实例的网络出入请求。建议按需开放和设置安全组出入规则时，不要默认开放所有的出入规则。更多详情，请参见[CreateSecurityGroup](#)。
- InstanceType: 实例规格。例如，选择1核2GiB n1.small则传入为ecs.n1.small。更多详情，请参见[实例规格族汇总](#)。
- ImageId: 镜像ID。您可以使用公共镜像或者自定义镜像。更多详情，请参见[DescribeImages](#)。

步骤1: 创建ECS实例

如以下代码所示，创建一台VPC类型的ECS实例，使用SSD云盘做系统盘，对应参数值为cloud_ssd，选择I/O优化实例则传入optimized属性。CreateInstance还提供了其他请求参数，您可以根据[CreateInstance文档](#)进行调整。

```
# create one after pay 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;
```

创建成功返回相应的实例ID和请求ID:

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

创建失败返回对应的错误码和错误信息:

```
{ "RequestId": "540CFF28-407A-40B5-B6A5-74Bxxxxxxxx", "HostId": "ecs.aliyuncs.com", "Code": "InvalidSecurityGroupId.NotFound", "Message": "The specified SecurityGroupId does not exist." }
```

步骤2: 查询ECS实例状态

有关ECS实例的状态信息，请参见[实例生命周期介绍](#)。其中:

- 仅Stopped状态的实例可以执行StartInstance操作。
- 仅Running状态的实例可以执行StopInstance操作。

查询ECS实例状态可以通过实例ID (InstanceId) 过滤。在DescribeInstances请求中通过传入JSON数组格式的String参数来查询实例状态。以下代码会检查实例的状态，并返回相应的实例信息：

```
# 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;
```

步骤3：启动ECS实例

创建成功后的ECS实例默认状态为Stopped。如果需要实例进入Running状态，可以通过StartInstance完成。

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

(可选) 步骤4：停止ECS实例

如果需要运行中的实例进入Stopped状态，可以通过StopInstance完成。

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

如何创建实例时自动检测并启动ECS实例

ECS实例的启动和停止都是异步操作，您可以在脚本创建并同时检测ECS实例符合状态时执行相应操作。以下示例代码的逻辑是：

1. 得到实例ID后，自动检测实例是否处于Stopped的状态。
2. 如果是Stopped，发起StartInstance指令。

3. 等待ECS实例进入Running状态。

```
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;
```

如何分配公网IP

如果在创建ECS实例的过程中，指定了公网带宽，若需要公网的访问权限请调用API来分配公网IP。更多详情，请参见[AllocatePublicIpAddress](#)。

创建包年包月ECS实例

通过API创建包年包月ECS实例的流程与ECS控制台创建流程不同，API流程使用的是自动扣费的模式。请在创建ECS实例之前确保账号支付方式有足够的余额或者信用额度，在创建ECS实例时实行直接扣费。

与创建按量付费ECS实例相比，您需要指定计费方式和计费时长。以下示例的时长为一个月。

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

创建包年包月实例的示例代码如下：

```
# 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_VSwitchId('vsw-vswitchid')
    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;
```

完整代码

本文的完整代码如下所示，您可以自行根据实际情况修改和设置参数取值。

```
# 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_t
ype=INSTANCE_TYPE,
                                           security_group_id=SECURITY_G
ROUP_ID)
    check_instance_running(instance_id=instance_id)
# create one after pay ecs instance.
def create_after_pay_instance(image_id, instance_type, security_g
roup_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_SystemDiskCategory('cloud_ssd')
    response = _send_request(request)
    instance_id = response.get('InstanceId')
    logging.info("instance %s created task submit successfully.",
instance_id)
```

```
    return instance_id;
# create one prepay ecs instance.
def create_prepay_instance(image_id, instance_type, security_group_id
):
    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_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_InstanceIds(json.dumps([instance_id]))
    response = _send_request(request)
    instance_detail = None
    if response is not None:
        instance_list = response.get('Instances').get('Instance')
        for item in instance_list:
            if item.get('Status') == status:
                instance_detail = item
                break;
    return instance_detail;
# send open api request
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
```

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

4.3 弹性管理ECS实例

您除了可以通过 ECS管理控制台创建或管理ECS实例外，您也能通过 API 管理或定制开发 ECS 实例。阿里云提供了 SDK 来包装API，将云服务器 ECS 的管理集成到已有系统中。本文基于 Python 的开发来说明如何通过API 管理 ECS 实例。如果您没有 Python 开发经验，也能通过本文完成云服务的开发。

获取 RAM 子账号 AK 密钥

使用API管理ECS实例，您需要能访问ECS资源的API密钥（AccessKey ID 和 AccessKey Secret）。为了保证云服务的安全，您需要创建一个能访问ECS资源的RAM用户，获取该用户的 AccessKey 密钥，并使用这个RAM用户和API管理ECS实例。

以下是获取RAM用户AccessKey密钥的操作步骤：

1. [创建RAM用户并获取AccessKey密钥](#)。
2. [为RAM用户授权](#)，授予RAM用户管理云服务器服务（ECS）的权限。

安装 ECS Python SDK

首先确保您已经具备Python的Runtime，本文中使用的Python版本为2.7+。

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

如果提示您没有权限，请切换sudo继续执行。

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

本文使用的SDK版本为2.1.2。

Hello Alibaba Cloud

创建文件 `hello_ecs_api.py`。为了使用SDK，首先实例化AcsClient对象，这里需要RAM用户的AccessKey ID和AccessKey Secret。



说明：

AccessKey ID和AccessKey Secret是RAM用户访问阿里云ECS服务API的密钥，具有该账户完全的权限，请妥善保管。

```
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')
```

完成实例化后可以进行第一个应用的开发。查询当前账号支持的地域列表。具体的文档参见 [查询可用地域列表](#)。

```
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()
```

在命令行运行 `python hello_ecs_api.py` 会得到当前支持的 Region 列表。类似的输出如下：

```
[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']
```

查询当前的 Region 下的 ECS 实例列表

查询实例列表和查询 Region 列表非常类似，替换入参对象为 `DescribeInstancesRequest` 即可，更多的查询参数参考 [查询实例列表](#)。

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

```
return instance_id
```

输出结果为如下:

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

更多的API参考 [ECS API 概览](#), 您可以尝试作一个 [查询磁盘列表](#), 将实例的参数替换为 DescribeDisksRequest。

完整代码示例

以上操作完整的代码示例如下所示。

```
# 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 aliynsdkecs.request.v20140526.DescribeInstancesRequest import
DescribeInstancesRequest
from aliynsdkecs.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')
        result = map(_print_instance_id, instance_list)
        logging.info("current region include instance %s", result)
def _print_instance_id(item):
    instance_id = item.get('InstanceId');
    return instance_id
def _print_region_id(item):
    region_id = item.get("RegionId")
    return region_id
```



```
# send open api request
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)
if __name__ == '__main__':
    logging.info("Hello Aliyun OpenApi!")
    hello_aliyun_regions()
    list_instances()
```

如您想了解 ECS 中 API 的其它操作，请参考 [ECS中的API操作](#)。

4.4 弹性释放ECS实例

云服务器 ECS 的一个重要特性就是按需创建资源。您可以在业务高峰期按需弹性地进行自定义资源创建，完成业务计算时释放资源。本篇将提供若干 Tips 帮助您更加便捷地完成云服务器的释放以及弹性设置。

本文将涉及到几个重要功能和相关API：

- [释放按量付费的云服务器](#)
- [设置按量付费实例的自动释放时间](#)
- [停止服务器](#)
- [查询实例列表](#)

释放后，实例所使用的物理资源将被回收，包括磁盘及快照，相关数据将全部丢失且永久不可恢复。如果您还想继续使用相关的数据，建议您释放云服务器之前一定要对磁盘数据做快照，下次创建 ECS 时可以直接通过快照创建资源。

释放云服务器

释放服务器，首先要求您的服务器处于停止状态。当服务器停止后，若影响到应用，您可以将服务器重新启动。

停止云服务器

停止服务器的指令非常简单，且对于按量付费和包年包月都是一样的。停止云服务器的一个参数是 ForceStop，若属性设置为 true，它将类似于断电，直接停止服务器，但不承诺数据能写到磁盘中。如果仅仅为了释放服务器，这个可以设置为 true。

```
def stop_instance(instance_id, force_stop=False):
    '''
    stop one ecs instance.
    :param instance_id: instance id of the ecs instance, like 'i-***'.
```

```
    :param force_stop: if force stop is true, it will force stop the
server and not ensure the data
write to disk correctly.
:return:
'''
    request = StopInstanceRequest()
    request.set_InstanceId(instance_id)
    request.set_ForceStop(force_stop)
    logging.info("Stop %s command submit successfully.", instance_id)
    _send_request(request)
```

释放云服务器

如果您没有停止服务器直接执行释放，可能会有如下报错：

```
{"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."}
```

当服务器处于Stopped状态时，您可以执行释放服务器。释放服务器的方法比较简单，参数如下：

- InstanceId: 实例的 ID。
- force: 如果将这个参数设置为 true，将会执行强制释放。即使云服务器不是Stopped状态也可以释放。执行的时候请务必小心，以防错误释放影响您的业务。

释放云服务器的Request如下：

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

释放云服务器成功的 Response 如下：

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

设置云服务器的自动释放时间

为了更加简化对云服务器的管理，您可以自定义云服务器的释放时间。当定时时间到后，阿里云将自动为您完成服务器的释放，无需手动执行释放。

**说明:**

自动释放时间按照 ISO8601 标准表示, 并需要使用 UTC 时间。格式为: yyyy-MM-ddTHH:mm:ssZ。如果秒不是 00, 则自动取为当前分钟开始时。自动释放的时间范围: 当前时间后 30 分钟 ~ 当前时间起 3 年。

```
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)
```

执行 `set_instance_auto_release_time('i-1111', '2017-01-30T00:00:00Z')` 后完成设置。

执行设置成功后, 您可以通过 `DescribeInstances` 来查询自动释放的时间设置。

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

```
return release_time
```

取消自动释放设置

如果您的业务有变化，需要取消自动释放设置。只需执行命令将自动释放时间设置为空即可。

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

完整代码



说明:

释放云服务器需谨慎。

```
# 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.ModifyInstanceAutoReleaseTimeR
equest import \
    ModifyInstanceAutoReleaseTimeRequest
from aliyunsdkecs.request.v20140526.StopInstanceRequest import
StopInstanceRequest
# configuration the log output formatter, if you want to save the
output to file,
# append ",filename='ecs_invoke.log'" after datefmt.
logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s %(filename)s[line:%(lineno)d]
%(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-***'.
```

```

: return:
'''
request = DescribeInstancesRequest()
request.set_InstanceIds(json.dumps([instance_id]))
response = _send_request(request)
if response is not None:
    instance_list = response.get('Instances').get('Instance')
    if len(instance_list) > 0:
        return instance_list[0]
def check_auto_release_time_ready(instance_id):
    detail = describe_instance_detail(instance_id=instance_id)
    if detail is not None:
        release_time = detail.get('AutoReleaseTime')
        return release_time
def release_instance(instance_id, force=False):
    '''
    delete instance according instance id, only support after pay
    instance.
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :param force:
    if force is false, you need to make the ecs instance stopped, you
    can
    execute the delete action.
    If force is true, you can delete the instance even the instance is
    running.
    :return:
    '''
    request = DeleteInstanceRequest();
    request.set_InstanceId(instance_id)
    request.set_Force(force)
    _send_request(request)
def set_instance_auto_release_time(instance_id, time_to_release = None
):
    '''
    setting instance auto delete time
    :param instance_id: instance id of the ecs instance, like 'i-***'.
    :param time_to_release: if the property is setting, such as '2017-
01-30T00:00:00Z'
    it means setting the instance to be release at that time.
    if the property is None, it means cancel the auto delete time.
    :return:
    '''
    request = ModifyInstanceAutoReleaseTimeRequest()
    request.set_InstanceId(instance_id)
    if time_to_release is not None:
        request.set_AutoReleaseTime(time_to_release)
    _send_request(request)
    release_time = check_auto_release_time_ready(instance_id)
    logging.info("Check instance %s auto release time setting is %s.
", instance_id, release_time)
def _send_request(request):
    '''
    send open api request
    :param request:
    :return:
    '''
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)

```

```
if __name__ == '__main__':
    logging.info("Release ecs instance by Aliyun OpenApi!")
    set_instance_auto_release_time('i-1111', '2017-01-28T06:00:00Z')
    # set_instance_auto_release_time('i-1111')
    # stop_instance('i-1111')
    # release_instance('i-1111')
    # release_instance('i-1111', True)
```

如您想了解 ECS 中 API 的其它操作，请参考 [ECS中的API操作](#)。

4.5 ECS实例续费

除了通过 ECS控制台或售卖页进行云服务器续费外，阿里云还支持直接通过 API 进行续费查询和续费管理

本文主要涉及如下关键功能：

- [按照过期时间查询云服务器](#)
- [续费实例](#)
- [查询云服务器自动续费时间](#)
- [设置云服务器自动续费时间](#)

对于包年包月的云服务器，生命周期非常重要。如果云服务器资源不能按时续费，将可能导致服务器被锁定甚至被释放，从而影响业务持续性。API 帮助您及时了解 and 检查资源的到期时间，并完成续费充值功能。

本篇需关注如下 API：

- [查询实例列表](#)
- [续费实例](#)

查询指定范围内到期的云服务器

查询实例列表的 API，通过过滤参数，您可以查询一定时间范围内到期的实例信息。通过设置过滤参数 `ExpiredStartTime` 和 `ExpiredEndTime`（时间参数按照 ISO8601 标准表示，并需要使用 UTC 时间。格式为：yyyy-MM-ddTHH:mmZ），可以方便地查询该时间范围内到期的实例列表。如果需要通过安全组进行过滤，只需加上安全组 ID 即可。

```
INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING = '2017-01-22T00:00Z'
INSTANCE_EXPIRE_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_g
roup_id=None):
    request = DescribeInstancesRequest()
    if check_need_renew is True:
        request.set_Filter3Key("ExpiredStartTime")
        request.set_Filter3Value(INSTANCE_EXPIRED_START_TIME_IN
_UTC_STRING)
        request.set_Filter4Key("ExpiredEndTime")
```

```
request.set_Filter4Value(INSTANCE_EXPIRE_END_TIME_IN_UT
C_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)
```

续费云服务器

续费实例只支持包年包月的服务器类型，不支持按量付费的服务器，同时要求用户必须支持账号的余额支付或信用支付。执行 API 的时候将执行同步的扣费和订单生成。因此，执行 API 的时候必须保证您的账号有足够的资金支持自动扣费。

```
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)
```

续费实例将会自动完成扣费。在完成续费后，您可以根据 InstanceId 查询实例的资源到期时间。由于 API 为异步任务，查询资源到期时间可能需要延迟 10 秒才会变化。

开启云服务器自动续费

为了减少您的资源到期维护成本，针对包年包月的 ECS 实例，阿里云还推出了 [自动续费功能](#)。自动续费扣款日为服务器到期前第 9 天的 08:00:00。如果前一日执行自动扣费失败，将会继续下一日定时执行，直到完成扣费或者 9 天后到期资源锁定。您只需要保证自己的账号余额或者信用额度充足即可。

· 查询自动续费设置

您可以通过 OpenAPI 来查询和设置自动续费。该 API 仅支持包年包月的实例，按量付费的实例执行将会报错。查询实例的自动续费状态支持一次最多查询 100 个包年包月的实例，多个实例 ID 以逗号连接。

DescribeInstanceAutoRenewAttribute 的入参为实例 ID。

InstanceId: 支持最多查询 100 个包年包月的实例，多个实例 ID 以逗号连接。

```
# check the instances is renew or not
def describe_auto_renew(instance_ids, expected_auto_renew=True):
    describe_request = DescribeInstanceAutoRenewAttributeRequest()
    describe_request.set_InstanceIds(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')

```

返回内容如下:

```

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

```

如果设置自动续费, 则返回的属性AutoRenewEnabled为 true, 否则返回 false。

· 设置和取消云服务器的自动续费

设置自动续费有三个入参:

- InstanceId: 支持最多查询100个包年包月的实例, 多个实例 ID 以逗号连接。
- Duration: 支持 1、2、3、6、12, 单位为月。
- AutoRenew: true/false, true为开启自动续费, false为取消自动续费。

```

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)

```

执行成功返回 Response 如下:

```

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

```

续费成功后, 您可以再执行一次查询。如果续费成功将返回续费时长以及是否开启自动续费。

```

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

```

完整代码

```

# 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 \
    DescribeInstanceAutoRenewAttributeRequest
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import \
    DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.ModifyInstanceAutoRenewAttributeRequest import \
    ModifyInstanceAutoRenewAttributeRequest
from aliyunsdkecs.request.v20140526.RenewInstanceRequest import \
    RenewInstanceRequest
logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s %(filename)s[line:%(lineno)d] \
%(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')
# 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_EXPIRE_END_TIME_IN_UTC_STRING = '2017-01-28T00:00Z'
def renew_job(page_size=100, page_number=1, check_need_renew=True, security_group_id=None):
    response = describe_need_renew_instance(page_size=page_size, page_number=page_number,
                                             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_id)
        logging.info("%s execute renew action ready", instance_ids)
def describe_need_renew_instance(page_size=100, page_number=1, instance_id=None,
                                 check_need_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_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.",
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
# 续费一个实例一个月
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 ', 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!")
    # 查询在指定的时间范围内是否有需要续费的实例。
    describe_need_renew_instance()
    # 续费一个实例，直接执行扣费
    renew_instance('i-1111')
    # 查询实例自动续费的状态
    # describe_instance_auto_renew_setting('i-1111,i-2222')
    # 设置实例自动续费

```

```
# setting_instance_auto_renew('i-1111,i-2222')
```

如您想了解 ECS 中 API 的其它操作，请参考 [ECS中的API操作](#)。

4.6 管理抢占式实例

本文介绍了如何使用阿里云ECS SDK合理快速地创建并管理抢占式实例。

准备工作

在执行操作之前，您需要：

- 了解能满足您业务要求的实例规格和地域。
- 熟悉了解阿里云ECS SDK的基础知识和调用方法。详细信息，请参见[SDK使用说明](#)。
- 抢占式实例代码需要依赖的ECS SDK版本4.2.0以上。以Java POM依赖为例，修改引入pom依赖：

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>3.2.8</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-ecs</artifactId>
  <version>4.2.0</version>
</dependency>
```

查询地域及可用的实例规格

使用[DescribeZones](#)查询可以创建抢占式实例的地域以及可用的实例规格。示例代码如下所示。

- [OpenApiCaller.java](#)

```
public class OpenApiCaller {
    IClientProfile profile;
    IAcsClient client;
    public OpenApiCaller() {
        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 (Throwable e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

- [DescribeZonesSample.java](#)

```
public class DescribeZonesSample {
```

```

public static void main(String[] args) {
    OpenApiCaller caller = new OpenApiCaller();
    DescribeZonesRequest request = new DescribeZonesRequest();
    request.setRegionId("cn-zhangjiakou");//可以通过DescribeRegionsRequest获取每个地域的RegionId
    request.setSpotStrategy("SpotWithPriceLimit");//对于查询是否可购买抢占式实例此项必填
    request.setInstanceChargeType("PostPaid");//后付费模式，抢占式实例必须是后付费模式
    DescribeZonesResponse response = caller.doAction(request);
    System.out.println(JSON.toJSONString(response));
}
}

```

以下为输出结果，可以查看每个地域可供选择的实例规格、磁盘类型、网络类型等信息。

```

{
  "requestId": "388D6321-E587-470C-8CFA-8985E2963DAE",
  "zones": [
    {
      "localName": "华北 3 可用区 A",
      "zoneId": "cn-zhangjiakou-a",
      "availableDiskCategories": [
        "cloud_ssd",
        "cloud_efficiency"
      ],
      "availableInstanceTypes": [
        "ecs.e4.large",
        "ecs.n4.4xlarge",
        "ecs.sn2.medium",
        "ecs.i1.2xlarge",
        "ecs.se1.2xlarge",
        "ecs.n4.xlarge",
        "ecs.se1ne.2xlarge",
        "ecs.se1.large",
        "ecs.sn2.xlarge",
        "ecs.se1ne.xlarge",
        "ecs.xn4.small",
        "ecs.sn2ne.4xlarge",
        "ecs.se1ne.4xlarge",
        "ecs.sn1.medium",
        "ecs.n4.8xlarge",
        "ecs.mn4.large",
        "ecs.e4.2xlarge",
        "ecs.mn4.2xlarge",
        "ecs.mn4.8xlarge",
        "ecs.n4.2xlarge",
        "ecs.e4.xlarge",
        "ecs.sn2ne.large",
        "ecs.sn2ne.xlarge",
        "ecs.sn1ne.large",
        "ecs.n4.large",
        "ecs.sn1.3xlarge",
        "ecs.e4.4xlarge",
        "ecs.sn1ne.2xlarge",
        "ecs.e4.small",
        "ecs.i1.4xlarge",
        "ecs.se1.4xlarge",
        "ecs.sn2ne.2xlarge",
        "ecs.sn2.3xlarge",
        "ecs.i1.xlarge",
        "ecs.n4.small",
        "ecs.sn1ne.4xlarge",

```

```

        "ecs.mn4.4xlarge",
        "ecs.sn1ne.xlarge",
        "ecs.se1ne.large",
        "ecs.sn2.large",
        "ecs.i1-c5d1.4xlarge",
        "ecs.sn1.xlarge",
        "ecs.sn1.large",
        "ecs.mn4.small",
        "ecs.mn4.xlarge",
        "ecs.se1.xlarge"
    ],
    "availableResourceCreation": [
        "VSwitch",
        "IoOptimized",
        "Instance",
        "Disk"
    ],
    "availableResources": [
        {
            "dataDiskCategories": [
                "cloud_ssd",
                "cloud_efficiency"
            ],
            "instanceGenerations": [
                "ecs-3",
                "ecs-2"
            ],
            "instanceTypeFamilies": [
                "ecs.mn4",
                "ecs.sn1",
                "ecs.sn2",
                "ecs.sn1ne",
                "ecs.xn4",
                "ecs.i1",
                "ecs.se1",
                "ecs.e4",
                "ecs.n4",
                "ecs.se1ne",
                "ecs.sn2ne"
            ],
            "instanceTypes": [
                "ecs.n4.4xlarge",
                "ecs.sn2.medium",
                "ecs.i1.2xlarge",
                "ecs.se1.2xlarge",
                "ecs.n4.xlarge",
                "ecs.se1ne.2xlarge",
                "ecs.se1.large",
                "ecs.sn2.xlarge",
                "ecs.se1ne.xlarge",
                "ecs.xn4.small",
                "ecs.sn2ne.4xlarge",
                "ecs.se1ne.4xlarge",
                "ecs.sn1.medium",
                "ecs.n4.8xlarge",
                "ecs.mn4.large",
                "ecs.mn4.2xlarge",
                "ecs.mn4.8xlarge",
                "ecs.n4.2xlarge",
                "ecs.sn2ne.large",
                "ecs.sn2ne.xlarge",
                "ecs.sn1ne.large",
                "ecs.n4.large",
                "ecs.sn1.3xlarge",
            ]
        }
    ]
}

```

```

        "ecs.sn1ne.2xlarge",
        "ecs.e4.small",
        "ecs.i1.4xlarge",
        "ecs.se1.4xlarge",
        "ecs.sn2ne.2xlarge",
        "ecs.sn2.3xlarge",
        "ecs.i1.xlarge",
        "ecs.n4.small",
        "ecs.sn1ne.4xlarge",
        "ecs.mn4.4xlarge",
        "ecs.sn1ne.xlarge",
        "ecs.se1ne.large",
        "ecs.sn2.large",
        "ecs.i1-c5d1.4xlarge",
        "ecs.sn1.xlarge",
        "ecs.sn1.large",
        "ecs.mn4.small",
        "ecs.mn4.xlarge",
        "ecs.se1.xlarge"
    ],
    "ioOptimized": true,
    "networkTypes": [
        "vpc"
    ],
    "systemDiskCategories": [
        "cloud_ssd",
        "cloud_efficiency"
    ]
}
],
"availableVolumeCategories": [
    "san_ssd",
    "san_efficiency"
]
}
]
}

```

查询抢占式实例的历史价格

使用 `DescribeSpotPriceHistory` 查询抢占式实例最近30天的价格变化数据，获得最佳性价比的地域和规格信息，示例代码（`DescribeSpotPriceHistorySample.java`）如下。

```

public class DescribeSpotPriceHistorySample {
    public static void main(String[] args) {
        OpenApiCaller caller = new OpenApiCaller();
        List<DescribeSpotPriceHistoryResponse.SpotPriceType> result =
        new ArrayList<DescribeSpotPriceHistoryResponse.SpotPriceType>();
        int offset = 0;
        while (true) {
            DescribeSpotPriceHistoryRequest request = new DescribeSpotPriceHistoryRequest();
            request.setRegionId("cn-hangzhou");//可以通过DescribeRegionsRequest获取可购买的每个地域的RegionId
            request.setZoneId("cn-hangzhou-b");//可用区必填
            request.setInstanceType("ecs.sn2.medium");//参考DescribeZones返回的实例类型，必填
            request.setNetworkType("vpc");//参考DescribeZones返回的网络类型，必填
            // request.setIoOptimized("optimized");//是否I/O优化类型，DescribeZones返回的IoOptimized，选填
        }
    }
}

```

```
// request.setStartTime("2017-09-20T08:45:08Z");//价格开始时间, 选填, 默认3天内数据
// request.setEndTime("2017-09-28T08:45:08Z");//价格结束时间, 选填
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 {
}
}
}
```

以下为返回结果示例。

```
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
```

重复以上步骤，您可以判断出该规格资源在可用区的价格变化趋势和最近价格。



说明:

您可以通过平均价格和最高价格来决定是否可以接受购买该抢占式实例，也可以通过更加合理的数据模型来分析历史价格数据，随时调整创建资源的规格和可用区，到达最佳性价比。

创建抢占式实例

在创建抢占式实例之前，您需要完成以下工作：

- 如果您使用自定义镜像创建抢占式实例，必须已经[CreateImage](#)。
- 在控制台[创建安全组](#)，或者使用[CreateSecurityGroup](#)创建安全组，并获取安全组ID（SecurityGroupId）。

- 在控制台创建VPC和交换机，或者使用CreateVpc和CreateVSwitch创建，并获取交换机ID（VSwitchId）。

使用CreateInstance创建抢占式实例。示例代码（CreateInstaneSample.java）如下。

```
public class CreateInstaneSample {
    public static void main(String[] args) {
        OpenApiCaller caller = new OpenApiCaller();
        CreateInstanceRequest request = new CreateInstanceRequest();
        request.setRegionId("cn-hangzhou");//地域ID
        request.setZoneId("cn-hangzhou-b");//可用区ID
        request.setSecurityGroupId("sg-bp11nhf94ivkdxwb2***");//提前创建的安全组ID
        request.setImageId("centos_7_03_64_20G_alibase_20170818.vhd");//建议选择您自己在该地域准备的自定义镜像
        request.setVSwitchId("vsw-bp164cyonthfudn9kj***");//VPC类型需要交换机ID
        request.setInstanceType("ecs.sn2.medium");//填入您询价后需要购买的规格
        request.setIoOptimized("optimized");//参见DescirbeZones返回参数
        request.setSystemDiskCategory("cloud_ssd");//参见DescirbeZones返回参数, 多选—cloud_ssd, cloud_efficiency, cloud
        request.setSystemDiskSize(40);
        request.setInstanceChargeType("PostPaid");//抢占式实例必须后付费
        request.setSpotStrategy("SpotWithPriceLimit");//SpotWithPriceLimit出价模式, SpotAsPriceGo不用出价, 最高按量付费价格
        request.setSpotPriceLimit(0.25F);//SpotWithPriceLimit出价模式生效, 您能接受的最高价格, 单位为元每小时, 必须高于当前的市场成交价才能成功
        CreateInstanceResponse response = caller.doAction(request);
        System.out.println(response.getInstanceId());
    }
}
```

回收抢占式实例

抢占式实例可能会因为价格因素或者市场供需变化而被强制回收，此时会触发抢占式实例的中断。释放前，抢占式实例会进入锁定状态，提示实例将会被自动回收。您可以针对实例回收状态自动化处理实例的退出逻辑。

目前，您可以通过以下任一种方式来获取抢占式实例的中断锁定状态：

- 通过实例元数据获取。运行以下命令：

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

如果返回为空，说明实例可持续使用。如果返回类似2015-01-05T18:02:00Z格式的信息（UTC时间），说明实例将于这个时间释放。

- 使用DescribeInstances，根据返回的OperationLocks判断实例是否进入待回收状态。代码示例（DescribeInstancesSample.java）如下。

```
public class DescribeInstancesSample {
    public static void main(String[] args) throws InterruptedException {

```



```

OpenApiCaller caller = new OpenApiCaller();
JSONArray allInstances = new JSONArray();
allInstances.addAll(Arrays.asList("i-bp18hgfai8ekoqwo0***", "i-
-bp1ecbyds24ij63w1***"));
while (!allInstances.isEmpty()) {
    DescribeInstancesRequest request = new DescribeIn
stancesRequest();
    request.setRegionId("cn-hangzhou");
    request.setInstanceIds(allInstances.toJSONString());//指定
实例ID, 效率最高
    DescribeInstancesResponse response = caller.doAction(
request);
    List<DescribeInstancesResponse.Instance> instanceList =
response.getInstances();
    if (instanceList != null && !instanceList.isEmpty()) {
        for (DescribeInstancesResponse.Instance instance :
instanceList) {
            System.out.println("result:instance:" + instance.
getInstanceId() + ",az:" + instance.getZoneId());
            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())) {
                        //do your action
                        System.out.println("spot instance will
be recycled immediately, instance id:" + instance.getInstanceId());
                        allInstances.remove(instance.
getInstanceId());
                    }
                }
            }
        }
        System.out.print("try describeInstances again later
...");
        Thread.sleep(2 * 60 * 1000);
    } else {
        break;
    }
}
}
}

```

触发回收时输出结果如下:

```

instance:i-bp1ecbyds24ij63w146c-->lockReason:Recycling,vmStatus:
Stopped
spot instance will be recycled immediately, instance id:i-bp1ecbyds2
4ij63w1***

```

其他操作

您还可以启动、停止、释放抢占式实例。具体的操作与按量付费实例没有区别。可以参见以下API文档:

- 启动实例: [StartInstance](#)

- 停止实例：[StopInstance](#)
- 释放实例：[DeleteInstance](#)

4.7 查询可用的变配资源

本文通过阿里云ECS Python SDK调用DescribeResourcesModification查询变更ECS实例配置时的可用规格资源列表，使用的Python语言为2.7版本。适用于作为升降配ECS实例规格操作的准备工作。

背景信息

升级或者降配云服务ECS配置时，一般通过[ModifyInstanceSpec](#)或[ModifyPrepayInstanceSpec](#)接口实现。为避免频繁报错，您可以在以下资源变配场景中通过[DescribeResourcesModification](#)接口查询变更ECS实例时的可用规格资源列表：

- 查询升级实例规格时的资源列表。
- 查询升级系统盘类型列表。

注意事项

[DescribeResourcesModification](#)可以设置MigrateAcrossZone=true查询跨系列更改实例配置的资源列表。跨系列更改实例配置会引起经典网络类型实例发生私网IP地址变化。如果您使用的是已停售的实例规格，请慎重从非I/O优化实例升级到I/O优化实例。根据实例网络类型的不同，将对实例信息造成以下影响：

| 变化项 | | 非I/O优化系列I实例跨规格变配 | | 其他系列实例规格变配 | |
|-----------------|--------------------------|------------------|---------|------------|---------|
| | | 经典网络 | 专有网络VPC | 经典网络 | 专有网络VPC |
| 私网IP地址 | | 发生变化 | 保持不变 | 发生变化 | 保持不变 |
| 磁盘设备名（仅Linux系统） | 普通云盘 cloud | 变更为xvda或者xvdb等 | | 保持不变 | |
| | 高效云盘 cloud_efficiency | 变更为vda或者vdb等 | | | |
| 软件授权码 | | 发生变化 | | | |

前提条件

请确保您已获取了AccessKey和资源所属的地域ID。更多详情，请参见[地域和可用区](#)和[创建AccessKey](#)。

安装Python SDK

以Linux操作系统安装Python SDK为例：

- 具有root权限的用户请执行：

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

- 普通用户请执行：

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



说明：

请确保您使用的Python SDK不低于4.6.3版本。更多详情，请前往[GitHub Repo Alibaba Cloud](#)获取最新版本ECS SDK。

使用场景示例

```
# 您的AccessKeyId
ak_id = "YOU_ACCESS_KEY_ID"

# 您的AccessKeySeret
ak_secret = "YOU_ACCESS_SECRET"

# 资源所在的地域ID
region_id = "cn-hangzhou"

# 选择的资源ID，如实例ID
resource_id = "i-xxxx"

# 目标资源类型，可选值：Instanceype| SystemDisk
destination_instance_type = 'InstanceType'
destination_system_disk = 'SystemDisk'

# 更改配置的操作类型，可选值：Upgrade | Downgrade | RenewDongrade |
RenewModify
operation_type = 'Upgrade'

# 目标实例规格，您可以通过DescribeInstanceTypes获得最新的规格列表。当参数
DestinationResource取值为SystemDisk，必须同时指定InstanceType参数
instance_type = "ecs.n1.large"

# 请慎用MigrateAcrossZone参数。MigrateAcrossZone (Boolean) 可以实现跨系列更
改配置，默认值：False
# 当参数MigrateAcrossZone取值为True时，一旦您根据返回信息升级了云服务器，请留意
以下注意事项：
# 经典网络类型实例：
# 1. 对于系列I实例，从非I/O优化实例升级到I/O优化实例时，实例私网IP地址、磁盘设备
名和软件授权码会发生变化。对于Linux实例，普通云盘（cloud）会被识别为xvda或者xvdb
等，高效云盘（cloud_efficiency）和SSD云盘（cloud_ssd）会被识别为vda或者vdb
等。
```

```
# 2. 对于其他系列实例，实例的私网IP地址会发生变化。
# VPC类型实例：对于系列I实例，从非I/O优化实例升级到I/O优化实例时，磁盘设备名和软件授权码会发生变化。Linux实例的普通云盘（cloud）会被识别为xvda或者xvdb等，高效云盘（cloud_efficiency和SSD云盘（cloud_ssd）会被识别为vda或者vdb等。
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"%(zoneId, values)

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

# 发起API请求
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
    except Exception as e:
```

```
logging.error(e)
```

场景1: 查询升级实例规格的资源列表

```
# 查询可用的实例规格变配列表
def describe_resource_instance_type():
    request = build_request()
    request.set_DestinationResource(destination_instance_type)
    _execute_request(request)
```

场景2: 查询升级系统盘规格或类型的资源列表

```
# 查询可用的系统盘变更配置列表
def describe_resource_system_disk():
    request = build_request()
    request.set_DestinationResource(destination_system_disk)
    request.set_InstanceType(instance_type)
    _execute_request(request)
```

完整代码

```
# coding=utf-8

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

import json
import logging

from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeResourcesModificationR
equest import DescribeResourcesModificationRequest

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')

# 您的AccessKeyId
ak_id = "YOU_ACCESS_KEY_ID"

# 您的AccessKeySecret
ak_secret = "YOU_ACCESS_SECRET"

# 资源所在的地域ID
region_id = "cn-hangzhou"

# 选择的资源ID, 如实例ID
resource_id = "i-xxxx"

# 目标资源类型, 可选值: InstanceType| SystemDisk
destination_instance_type = 'InstanceType'
destination_system_disk = 'SystemDisk'

# 更改配置的操作类型, 可选值: Upgrade | Downgrade | RenewDowngrade |
RenewModify
operation_type = 'Upgrade'
```

```

# 目标实例规格，您可以通过DescribeInstanceTypes获得最新的规格列表。当参数
DestinationResource取值为SystemDisk，必须同时指定InstanceType参数
instance_type = "ecs.n1.large"

# 请慎用MigrateAcrossZone参数。MigrateAcrossZone (Boolean) 可以实现跨系列更
改配置，默认值: False
# 当参数MigrateAcrossZone取值为True时，一旦您根据返回信息升级了云服务器，请注意
以下注意事项：
# 经典网络类型实例：
# 1. 对于系列I实例，从非I/O优化实例升级到I/O优化实例时，实例私网IP地址、磁盘设备
名和软件授权码会发生变化。对于Linux实例，普通云盘 (cloud) 会被识别为xvda或者xvdb
等，高效云盘 (cloud_efficiency) 和SSD云盘 (cloud_ssd) 会被识别为vda或者vdb
等。
# 2. 对于其他系列实例，实例的私网IP地址会发生变化。
# 专有网络VPC类型实例：对于系列I实例，从非I/O优化实例升级到I/O优化实例时，磁盘设
备名和软件授权码会发生变化。Linux实例的普通云盘 (cloud) 会被识别为xvda或者xvdb
等，高效云盘 (cloud_efficiency) 和SSD云盘 (cloud_ssd) 会被识别为vda或者vdb
等。
migrate_across_zone = False

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

# 查询升级实例规格的资源列表
def describe_resource_instance_type():
    request = build_request()
    request.set_DestinationResource(destination_instance_type)
    _execute_request(request)

# 查询升级系统盘规格或类型的资源列表
def describe_resource_system_disk():
    request = build_request()
    request.set_DestinationResource(destination_system_disk)
    request.set_InstanceType(instance_type)
    _execute_request(request)

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

```

```
        value = supportedResource.get('Value')
        values.append(value)
    print "ecs in zone %s resource value list is %s"%(zoneId
, values)

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

# 发起API请求
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()
```