Alibaba Cloud
Elastic Compute Service

SDK レファレンス

Document Version20190629

目次

1 SDK リファレンス

現在、Alibaba Cloud の ECS では次の SDK プログラミング言語を提供しています。

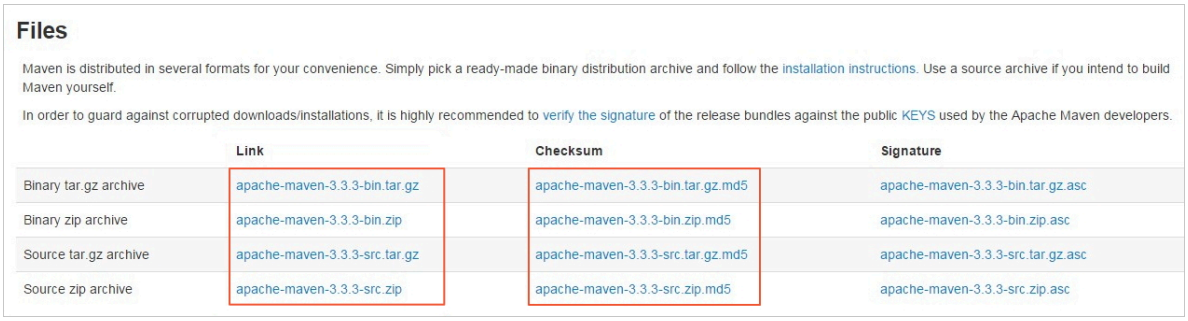
- *Java*
- *Python*
- *PHP*
- *C++*
- *. NET*

2 Java SDK のダウンロードとインストール

従来のバージョンの Java SDK は、スタンドアロンダウンロードパッケージとしてのご利用が可能でした。 Alibaba Cloud Java SDK の新しいバージョンの全てが、管理を容易にするために Maven リポジトリに保管されます。

たとえば、Eclipse Luna を 64 ビット版の Windows 7 でご利用になる際には、次の手順に従って Java SDK をダウンロードします。

1. 『*Maven の公式ダウンロードページ*』にアクセスして、お使いのオペレーティングシステムに対応する Maven ソフトウェアをダウンロードします。 Checksum ファイルをチェックすることによって、ダウンロードしたファイルを確認することができます。



Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

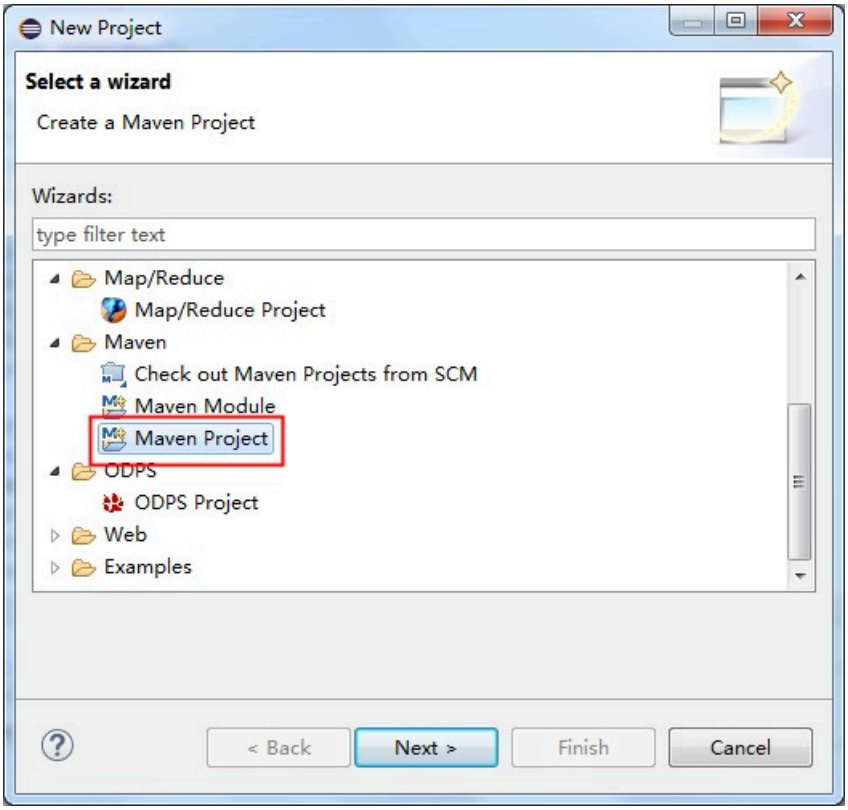
	Link	Checksum	Signature
Binary tar.gz archive	apache-maven-3.3.3-bin.tar.gz	apache-maven-3.3.3-bin.tar.gz.md5	apache-maven-3.3.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.3.3-bin.zip	apache-maven-3.3.3-bin.zip.md5	apache-maven-3.3.3-bin.zip.asc
Source tar.gz archive	apache-maven-3.3.3-src.tar.gz	apache-maven-3.3.3-src.tar.gz.md5	apache-maven-3.3.3-src.tar.gz.asc
Source zip archive	apache-maven-3.3.3-src.zip	apache-maven-3.3.3-src.zip.md5	apache-maven-3.3.3-src.zip.asc

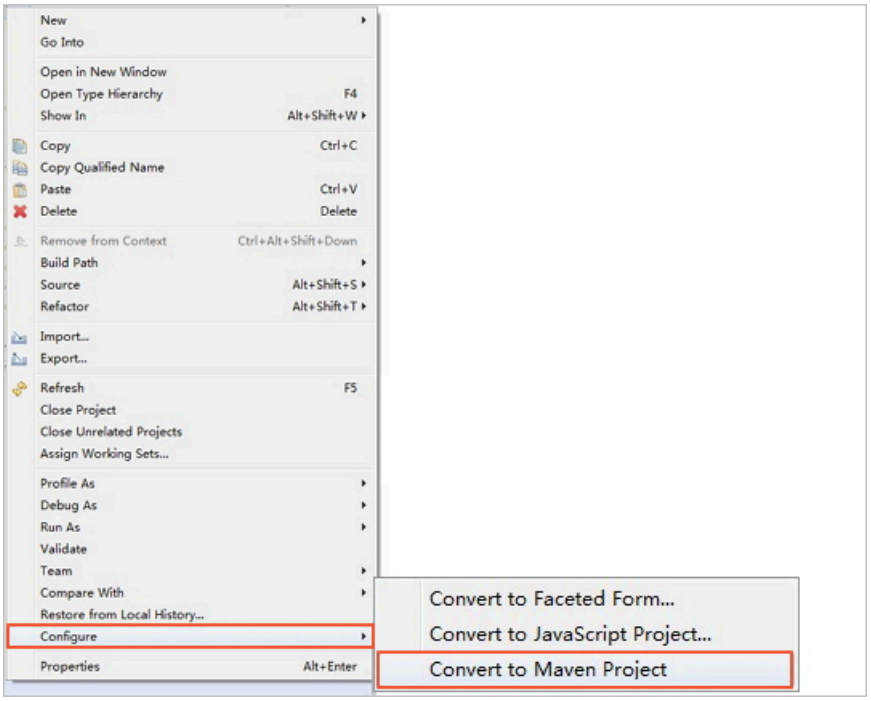
2. 『*Java SDK の ダウンロードページ*』にアクセスし、Alibaba Cloud の SDK が格納されている Maven リポジトリを Maven ソフトウェアに追加します。ダウンロードした

Maven パッケージを解凍し、Alibaba Cloud Maven リポジトリを conf ディレクトリ中の "settings.xml" ファイルに追加します。

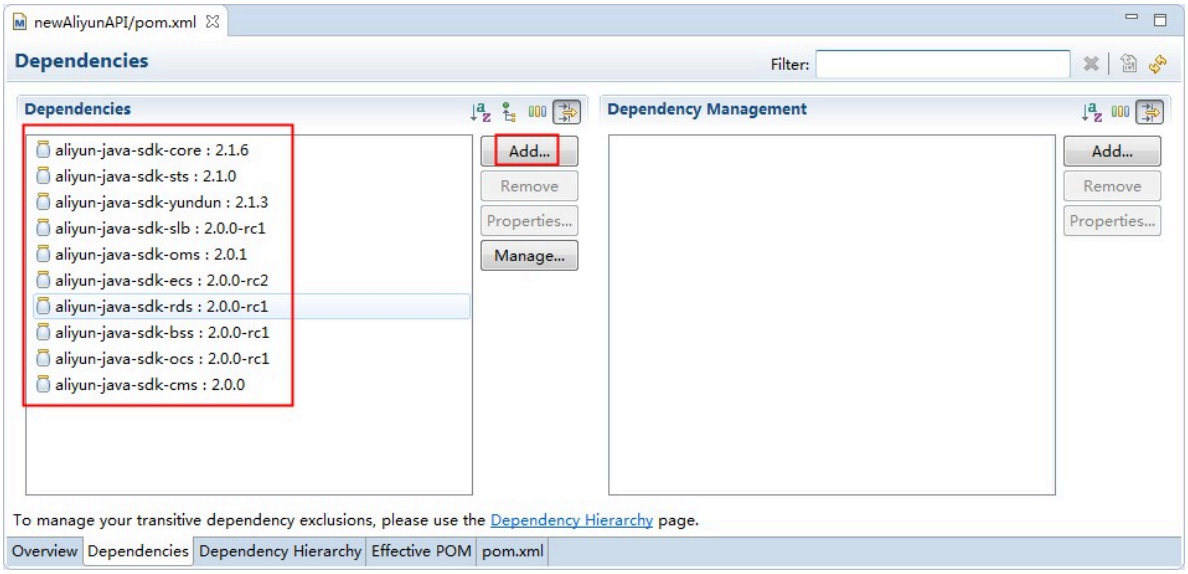
```
238          |  </configuration>
239          | </plugin>
240          | ...
241          |
242          | NOTE: If you just wanted to inject this configuration whenever someone set 'target-env' to
243          |       anything, you could just leave off the <value/> inside the activation-property.
244          |
245          -->
246        <profile>
247          <id>nexus</id>
248
249          <repositories>
250        <repository>
251            <id>sonatype-nexus-staging</id>
252            <name>Sonatype Nexus Staging</name>
253            <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
254            <releases>
255                <enabled>true</enabled>
256            </releases>
257            <snapshots>
258                <enabled>true</enabled>
259            </snapshots>
260        </repository>
261    </repositories>
262
```

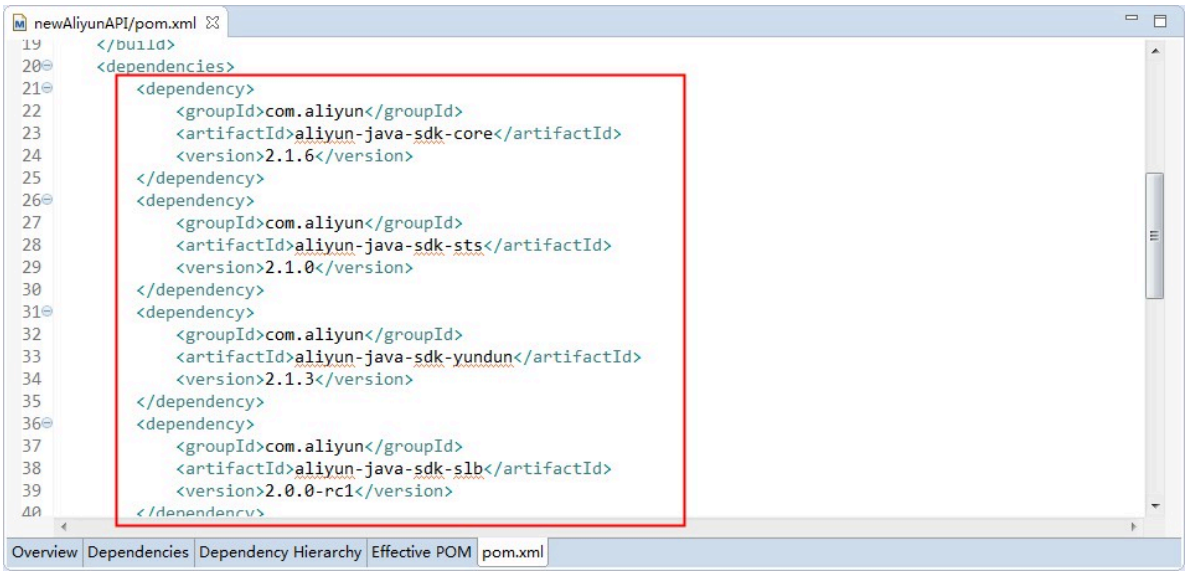
3. **Eclipse で Maven プロジェクトを作成する、もしくは既存のプロジェクトを Maven プロジェクトに変換します。**



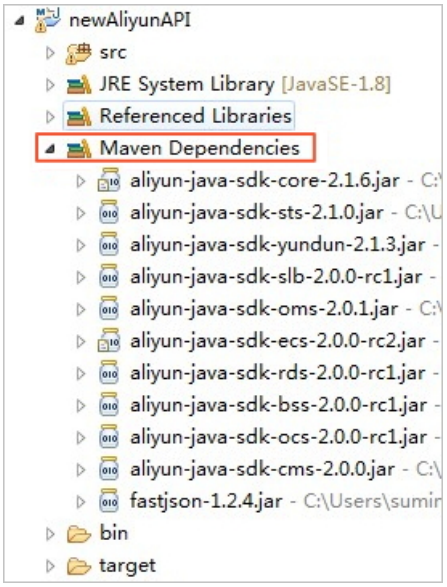


4. プロジェクトの下の "pom.xml" ファイルを開き、グラフィカルインタフェースで Maven の依存関係を追加する、もしくは "pom.xml" ファイルで依存関係を編集します。





5. 設定を保存します。 Alibaba Cloud SDK の JAR パッケージは、Maven の依存関係へとロードされます。



SDK のバージョンを確認する方法

従来の SDK と新しい SDK の違いは、次の表でハイライトしています。

比較対象	従来 SDK	最新 SDK
操作のリクエスト方法	execute()	getAcsResponse()
AccessKey と AccessKeySecret を保持するクラス	AliyunClient	IClientProfile
ストレージ資格情報オブジェクトを生成する方法	new DefaultAliyunClient(APIUrl, Access Key, Access Key Secret)	DefaultProfile.getProfile(RegionId, Access Key, Access Key Secret)
パッケージのプレフィックス	com.aliyun.api	com.aliyuncs

現行バージョンの SDK を使用している場合は、高度な機能を利用するために最新バージョンに切り替えることを推奨します。

3 ECS SDK を使用した例

新しいバージョンの SDK ファイル名は、通常、"aliyun-XXXX-sdk" で始まり、次に製品名が続きます。たとえば "aliyun-java-sdk-ecs" のようなパッケージ名です。 "aliyun-java-sdk-core" コアパッケージは、"IClientProfile"、"IAcsClient"、および例外クラスなど、すべての SDK 製品で使用されている特定のクラスをカプセル化します。 クラスは製品ごとに異なる JAR パッケージに格納されています。

「*AccessKey* 」の作成ができる状態にします。

Java SDK を使用した例

この例では、API メソッド*DescribeImages*は使用可能なイメージリソースを照会するために使用されます。 API メソッドを Java SDK を使用するプロセスの例として説明します。 "aliyun-java-sdk-core" パッケージには、"IClientProfile" クラスと "IAcsClient" クラスが含まれ、"aliyun-java-sdk-ecs" パッケージにはそれ以外のクラスが含まれています。

1. プロファイルオブジェクトの作成: "profile" という名前の "IClientProfile" クラスのインスタンスを作成します。このインスタンスには、"AccessKeyID"、"AccessKeySecret"、および " cn - hangzhou " などのデフォルトのリージョン情報が含まれています。 Alibaba Cloud のリージョン詳細については、「リージョンとゾーン」をご参照ください。

   ```
   IClientPro file profile = DefaultPro file . getProfile (" cn
   - hangzhou ", ak , aks ); # ak is your AccessKey , and
   aks is your AccessKeyS ecret .
   ```

2. クライアントオブジェクトの作成: 上記で作成した "IClientProfile" プロファイルから "client" という名前の "IAcsClient" オブジェクトを作成し、その後のレスポンスは "IClientProfile" から取得します。 このオブジェクトには、後で取得するレスポンスデータが含まれています。

   ```
   IAcsClient client = new DefaultAcs Client ( profile );
   ```

3. Request クラスの作成: メソッドに対応するリクエストクラスを作成し、API メソッド名の最後に " Request " を追加してクラスに名前を付けます。 たとえば、イメージリストを照会する API メソッドの名前が*DescribeImages*である場合、対応するリクエストクラス名は

"DescribeImagesRequest" です。 コンストラクターを使用して、デフォルトの "Describe" クラスを生成します。

```
DescribeIm  agesReques  t  describe  =  new  DescribeIm
agesReques  t ();
```

4. リクエストパラメーターの指定: セッター " setXxx " を使用して必要な API パラメーターを指定します。 たとえば、*DescribeImages*API メソッドは、"IClientProfile" に既にリージョン情報が含まれているため、オプションである " RegionId " パラメーターを必要とします。 他のパラメーターを設定するには、他の " setters " を使用します。 たとえば、カスタムイメージを照会するには、 " ImageOwner Alias " を " self " に設定します。

```
describe . setImageOw  nerAlias (" self ");
```

5. "IAcsClient" オブジェクトを使用してリクエストに対するレスポンスを取得します。

```
DescribeIm  agesRespon  se  response  =  client . getAcsResp  onse
( describe );
```

6. レスポンスからのレスポンスパラメーター取得: レスポンスでゲッター " getXxx " を呼び出し、戻り値 "ImageName" などを取得します。 API メソッドごとに、戻り値に構造化情報が含まれることがあります。 たとえば、"DescribeImages" のメソッドでは、戻り値には Java SDK のイメージコレクションが含まれます。 " getImages ()" メソッドを呼び出してイメージオブジェクトリストを照会し、リストを繰り返してイメージを照会します。次に、" getXxx " を呼び出し、さらに情報を取得します。

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

これで呼び出し処理の完了です。

PHP SDK に関する注意事項

PHP SDK のを使用方法は、Java SDK と同じ方法です。 以下の手順に従います。

1. プロファイルオブジェクトを作成します。
2. クライアントオブジェクトを作成します。
3. リクエストオブジェクトを作成します。
4. リクエストパラメーターを設定します。
5. リクエストオブジェクトを渡して戻り値を取得することにより、クライアントオブジェクトに対応する API メソッドを呼び出します。

6. レスポンスの戻り値を取得します。

Python SDK に関する注意事項

Python SDK を使用する場合、プロファイルを作成する必要はありません。 代わりにクライアントを直接作成し、残りの手順を続行します。

参考文献

- ECS での利用可能な API メソッドの詳細については、*API Overview*をご参照ください。
- "AccessKey" を作成する方法の詳細については、「*AccessKey の作成*」をご参照ください。

4 API の使用による ECS の実行

4.1 複数インスタンスの一括作成

"RunInstances" を使用して一括で複数の ECS インスタンスを作成できます。 アプリケーションの開発と展開を迅速に行うのに役立ちます。

*CreateInstance*と比較して、*RunInstances*には次のような利点があります。

- "RunInstances" には、一度のリクエストで、最大 100 個のインスタンスまたは プリエンプティブインスタンスを作成し、自動的に実行するための "Amount" が含まれています 。
- 一つのインスタンスを作成した際に、そのインスタンスのステータスは自動的に "Starting" に変わり、その後 "Running" に変わります。 "StartInstance" 操作を呼び出す必要はありません。
- " InternetMa xBandwidth Out " の値を 0 より大きく設定した場合は、インスタンスにインターネット IP が割り当てられます。
- すべてのリクエストを満たすため、 100 個の「プリエンプティブインスタンス」を一度に作成することもできます。
- " AutoReleas eTime " の設定で、リリースプランをスケジュールすることができ、作成したパラメーターの数を " Amount " で設定できます。 エラーコードと "RunInstances" の使用可能なパラメーターは、"CreateInstance" と完全に互換性があります。
- " InstanceId Sets " はリクエスト後に、すべての " InstanceId s " をリスト化するため、「作成されたインスタンスのステータスポーリング」が許可されます。

前提条件

"AccessKey" が作成されていること。

:

プライマリアカウントの "AccessKey" を使用しないでください。 ライマリアカウントの "AccessKey" が開示されている場合、すべてのリソースが安全でない可能性があります。 RAM ユーザーアカウントの "AccessKey" を使用すれば、"AccessKey" が漏洩するリスクを軽減できます。

ECS Python SDK のインストール

Python のランタイムを所有していることをご確認ください。 ここでは、Python 2.7 以降のバージョンを例として取り上げております。

*SDK のバージョン*は 4.4.3 です。

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

操作許可がないことを示すメッセージを受け取った場合は、" sudo " に切り替えます。

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

本説明で使用している SDK のバージョンは 4.4.3 です。 旧バージョンをお使いの場合、更新することを推奨します。

インスタンスの作成

"RunInstancesRequest" オブジェクトを作成し、関連パラメーターを入力します。

この例では、2 つのインスタンスを作成し、10 秒ごとにインスタンスのステータスを自動的にチェックするよう指定します。 インスタンスのステータスが "Running" に変わると、作成は完了します。

```
# your access key Id
ak_id = "YOU_ACCESS_KEY_ID"
# your access key secret
ak_secret = "YOU_ACCESS_SECRET"
region_id = "cn-beijing"
# your expected instance type
instance_type = "ecs.n4.small"
# The selected vswitchId
vswitch_id = "vws-xxxxx"
# The selected image info
image_id = "centos_7_0_3_64_20G_alibase_201_70818.vhd"
# The selected security group of VPC network
security_group_id = "sg-xxxxx"
# instance number to launch, support 1-100, default
value is 100
amount = 2;
# The auto release time is in accordance with ISO8601
and must be UTC. The format is `yyyy-MM-ddTHH
:mm:ssZ`. The release time must be at least 30
minutes later than the current time and less than 3
years from the current time.
auto_release_time = "2017-12-05T22:40:00Z"
clt = client.AcsClient(ak_id, ak_secret, 'cn-beijing')
# create instance automatic running
def batch_create_instance():
    request = build_request()
    request.set_Amount(amount)
    _execute_request(request)
def _execute_request(request):
    response = _send_request(request)
    if response.get('Code') is None:
```

```
        instance_i  ds  =  response . get (' InstanceId  Sets '). get
 (' InstanceId  Set ')
        running_am  ount  =  0
        while   running_am  ount  <  amount :
            time . sleep ( 10 )
            running_am  ount  =  check_inst  ance_runni  ng (
 instance_i  ds )
    print (" ecs   instance  % s   is   running ",  instance_i  ds )
 def   check_inst  ance_runni  ng ( instance_i  ds ):
    request  =  DescribeIn  stancesReq  uest ()
    request . set_Instan  ceIds ( json . dumps ( instance_i  ds ))
    response  =  _send_requ  est ( request )
    if   response . get (' Code ')  is   None :
        instances_  list  =  response . get (' Instances '). get ('
 Instance ')
        running_co  unt  =  0
        for   instance_d  etail   in   instances_  list :
            if   instance_d  etail . get (' Status ') == " Running ":
                running_co  unt  +=  1
        return   running_co  unt
 def   build_requ  est ():
    request  =  RunInstanc  esRequest ()
    request . set_ImageI  d ( image_id )
    request . set_VSwitc  hId ( vswitch_id )
    request . set_Securi  tyGroupId ( security_g  roup_id )
    request . set_Instan  ceName (" Instance12 - 04 ")
    request . set_Instan  ceType ( instance_t  ype )
    return   request
#  send   open   api   request
 def   _send_requ  est ( request ):
    request . set_accept  _format (' json ')
    try :
        response_s  tr  =  clt . do_action ( request )
        logging . info ( response_s  tr )
        response_d  etail  =  json . loads ( response_s  tr )
        return   response_d  etail
    except   Exception   as   e :
        logging . error ( e )
```

インターネット IP を用いたインスタンスの作成

インスタンスを作成し、属性の行に追加してインターネット帯域幅を指定します。 この例では、各インスタンスに 1 Mbit/s の帯域幅を割り当てます。

```
#  create   instance   with   public   ip .
 def   batch_crea  te_instanc  e_with_pub  lic_ip ():
    request  =  build_requ  est ()
    request . set_Amount ( amount )
    request . set_Intern  etMaxBandw  idthOut ( 1 )
    _execute_r  equest ( request )
```

自動リリース時間を用いたインスタンスの作成

「インスタンスを作成」し、インスタンスの自動リリース時間を指定する属性行を追加します。 自動リリース時間は「 *ISO8601* 」に準拠しており、UTC でなければなりません。 形式は "

`YYYY-MM-DDTHH:mm:ssZ"` です。リリース時間は、現在の時刻より 30 分早く設定したり、3 年を超えて設定することはできません。

```
# create instance with auto release time.
def batch_crea te_instanc e_with_aut o_release_ time():
    request = build_requ est()
    request.set_Amount(amount)
    request.set_AutoRe leaseTime(auto_relea se_time)
    _execute_r equest(request)
```

完成コード例

完成コード例は以下のとおりです。

```
# 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 aliyunsdkc ore import client
from aliyunsdke cs.request.v20140526. DescribeIn
stancesReq uest import DescribeIn stancesReq uest
from aliyunsdke cs.request.v20140526. RunInstanc esRequest
  import RunInstanc esRequest
logging.basicConfi g(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_t ype = "ecs.n4.small"
# The selected vswitchId.
vswitch_id = "vws-xxxxx"
# The selected image info
image_id = "centos_7_0 3_64_20G_a libase_201 70818.vhd"
# The selected security group of VPC network
security_g roup_id = "sg-xxxxx"
# instance number to launch, support 1-100, default
value is 100
amount = 2;
# The auto release time is in accordance with ISO8601
  and must be UTC. The format is `YYYY-MM-DDTHH
:mm:ssZ`. The release time must be at least 30
minutes later than the current time and less than 3
  years from the current time.
auto_relea se_time = "2017-12-05T22:40:00Z"
clt = client.AcsClient(ak_id, ak_secret, 'cn-beijing')
# create instance automatic running
def batch_crea te_instanc e():
    request = build_requ est()
    request.set_Amount(amount)
```

```
    _execute_r  equest ( request )
#  create   instance   with   public   ip .
 def   batch_crea  te_instanc  e_with_pub  lic_ip ():
    request  =  build_requ  est ()
    request . set_Amount ( amount )
    request . set_Intern  etMaxBandw  idthOut ( 1 )
    _execute_r  equest ( request )
#  create   instance   with   auto   release   time .
 def   batch_crea  te_instanc  e_with_aut  o_release_  time ():
    request  =  build_requ  est ()
    request . set_Amount ( amount )
    request . set_AutoRe  leaseTime ( auto_relea  se_time )
    _execute_r  equest ( request )
 def   _execute_r  equest ( request ):
    response  =  _send_requ  est ( request )
    if   response . get (' Code ')  is   None :
        instance_i  ds  =  response . get (' InstanceId  Sets '). get
 (' InstanceId  Set ')
        running_am  ount  =  0
        while   running_am  ount  <  amount :
            time . sleep ( 10 )
            running_am  ount  =  check_inst  ance_runni  ng (
 instance_i  ds )
    print (" ecs   instance  % s   is   running ",  instance_i  ds )
 def   check_inst  ance_runni  ng ( instance_i  ds ):
    request  =  DescribeIn  stancesReq  uest ()
    request . set_Instan  ceIds ( json . dumps ( instance_i  ds ))
    response  =  _send_requ  est ( request )
    if   response . get (' Code ')  is   None :
        instances_  list  =  response . get (' Instances '). get ('
 Instance ')
        running_co  unt  =  0
        for   instance_d  etail   in   instances_  list :
            if   instance_d  etail . get (' Status ') == " Running ":
                running_co  unt  +=  1
        return   running_co  unt
 def   build_requ  est ():
    request  =  RunInstanc  esRequest ()
    request . set_ImageI  d ( image_id )
    request . set_VSwitc  hId ( vswitch_id )
    request . set_Securi  tyGroupId ( security_g  roup_id )
    request . set_Instan  ceName (" Instance12 - 04 ")
    request . set_Instan  ceType ( instance_t  ype )
    return   request
#  send   open   api   request
 def   _send_requ  est ( request ):
    request . set_accept  _format (' json ')
    try :
        response_s  tr  =  clt . do_action ( request )
        logging . info ( response_s  tr )
        response_d  etail  =  json . loads ( response_s  tr )
        return   response_d  etail
    except   Exception   as   e :
        logging . error ( e )
 if   __name__  == ' __main__ ':
    print  " hello   ecs   batch   create   instance "
   #  batch_crea  te_instanc  e ()
   #  batch_crea  te_instanc  e_with_pub  lic_ip ()
```

```
    #  batch_crea  te_instanc  e_with_aut  o_release_  time ()
```

4.2 インスタンスの作成

ECS コンソールや購入ページに加え、API コードを使用して ECS インスタンスを自由に作成したり管理することができます。 ここでは、Python を使って ECS インスタンスを作成する方法について説明します。

ECS インスタンスを作成する際には、次の API に注目します。

- *CreateInstance*
- *DescribeInstances*
- *StartInstance*
- *AllocatePublicIpAddress*

従量課金 ECS インスタンスの作成

- 必須属性
  - SecurityGroupId: セキュリティグループ ID。 セキュリティグループは、インスタンスのネットワークアクセスリクエストを保護するため、ファイアウォールルールに基づいてインスタンスの設定をするために使用されます。 セキュリティグループのアクセスルールを設定する際は、すべてのアクセスルールではなく、必要なアクセスルールのみ有効にすることを推奨します。 ECS コンソールでセキュリティグループを作成します。
  - InstanceType: インスタンスタイプ。 「*ECS 購入ページ*」をご参照ください。 オプションの“one-core 2GB n1.small”は、入力パラメーターが“ecs.n1.small”であることを表しています。
  - ImageId: イメージ ID。 「*ECS コンソール*」のイメージリストをご参照ください。 パブリックイメージやカスタムイメージをフィルタリングします。

  その他パラメーター設定については、「*CreateInstance*」をご参照ください。
- ECS インスタンスの作成

  次のコードは、SSD をシステムディスク、“cloud_ssd”をディスクパラメーターとして、I/O に最適化されたクラシックネットワーク ECS インスタンスの作成方法を紹介しています。

```
#  create   one   after   pay   ecs   instance .
 def   create_aft  er_pay_ins  tance ( image_id ,  instance_t  ype
 ,  security_g  roup_id ):
     request  =  CreateInst  anceReques  t ();
     request . set_ImageI  d ( image_id )
     request . set_Securi  tyGroupId ( security_g  roup_id )
     request . set_Instan  ceType ( instance_t  ype )
     request . set_IoOpti  mized (' optimized ')
```

```
    request . set_System  DiskCatego  ry (' cloud_ssd ')
    response  =  _send_requ  est ( request )
    instance_i  d  =  response . get (' InstanceId ')
    logging . info (" instance  % s  created  task  submit
successful  ly .",  instance_i  d )
    return  instance_i  d ;
```

ECS インスタンスが正常に作成されると、インスタンス ID が返されます。 作成に失敗した場合は、エラーコードが返されます。 多くのパラメーターがあるため、*ECS* 購入ページで調整を行うことができます。

```
{" InstanceId ":" i -***"," RequestId ":" 006C1303 - BAC5 - 48E5 -
 BCDF - 7FD5C2E639  5D "}
```

- ECS ライフサイクル

  異なる ECS ステータスの操作については、「ECS」 インスタンスのライフサイクル をご参照ください。

  インスタンスはステータスが "停止済" のときにのみ、スタート操作を行うことができ、ステータスが "実行中" のときのみ、"停止" 操作を行うことができます。 ECS の状態を照会するためには、パラメーターのインスタンス ID を入力すると、インスタンスリストをフィルタリングできます。 "DescribeInstancesRequest" を呼び出す場合、JSON の文字配列を入力して、リソースの状態を照会します。 単一のインスタンスの状態を照会する場合、"DescribeInstanceAttribute" ではなく "DescribeInstances" を使用することを推奨します。前者の API のほうが多くの属性やコンテンツを返すからです。

  次のコードは、インスタンスの状態を確認する際に使用します。 システムは、インスタンスのステータスが入力パラメーターに適合した場合にのみ、インスタンスの詳細を返します。

```
#  output  the  instance  owned  in  current  region .
 def  get_instan  ce_detail_  by_id ( instance_i  d ,  status ='
 Stopped '):
    logging . info (" Check  instance  % s  status  is  % s ",
 instance_i  d ,  status )
    request  =  DescribeIn  stancesReq  uest ()
    request . set_Instan  ceIds ( json . dumps ([ instance_i  d ]))
    response  =  _send_requ  est ( request )
    instance_d  etail  =  None
    if  response  is  not  None :
        instance_l  ist  =  response . get (' Instances '). get ('
 Instance ')
        for  item  in  instance_l  ist :
            if  item . get (' Status ') ==  status :
                instance_d  etail  =  item
                break ;
```

```
        return   instance_d  etail ;
```

- ECS インスタンスの開始

  ECS インスタンスが正常に作成されると、デフォルトのインスタンスは "停止済" のステータスになります。 ステータスを "実行中" に変更するには、Start コマンドを送信します。

```
def   start_inst  ance ( instance_i  d ):
    request  =  StartInsta  nceRequest ()
    request . set_Instan  ceId ( instance_i  d )
    _send_requ  est ( request )
```

- ECS インスタンスの停止

  ECS インスタンスを停止するには、入力インスタンス ID を使用します。

```
def   stop_insta  nce ( instance_i  d ):
    request  =  StopInstan  ceRequest ()
    request . set_Instan  ceId ( instance_i  d )
    _send_requ  est ( request )
```

- ECS インスタンス作成時の “ ECS 自動起動 ” の有効方法

  ECS の開始と停止操作は非同期で行われます。 スクリプトが ECS インスタンスを作成し、適切な状態にあるかどうかを検出している際に、操作を行うことができます。

  正常に作成された ECS インスタンス ID を取得した後、インスタンスのステータスが "停止済" かどうかをチェックします。 ステータスが "停止済" の場合は、ECS コマンドの `Start` を送信し、ECS のステータスが "実行中" に変わるまで待機します。

```
def   check_inst  ance_runni  ng ( instance_i  d ):
    detail  =  get_instan  ce_detail_  by_id ( instance_i  d =
instance_i  d ,  status = INSTANCE_R  UNNING )
    index  =  0
    while   detail   is   None   and   index  & lt ;  60 :
        detail  =  get_instan  ce_detail_  by_id ( instance_i  d =
instance_i  d );
        time . sleep ( 10 )
    if   detail   and   detail . get (' Status ') == ' Stopped ':
        logging . info (" instance  % s   is   stopped   now .")
        start_inst  ance ( instance_i  d = instance_i  d )
        logging . info (" start   instance  % s   job   submit .")
    detail  =  get_instan  ce_detail_  by_id ( instance_i  d =
instance_i  d ,  status = INSTANCE_R  UNNING )
    while   detail   is   None   and   index  & lt ;  60 :
        detail  =  get_instan  ce_detail_  by_id ( instance_i  d =
instance_i  d ,  status = INSTANCE_R  UNNING );
        time . sleep ( 10 )
    logging . info (" instance  % s   is   running   now .",
instance_i  d )
     return   instance_i  d ;
```

- パブリック IP アドレスの割り当て

  ECS インスタンスを作成する際に、パブリックネットワークの帯域幅を指定する場合は、パブリックネットワークアクセスに対するインスタンスにパブリック IP アドレスを割り当てるた

めに API を呼び出す必要があります。詳しくは、「*AllocatePublicIpAddress*」をご参照ください。

サブスクリプションモードによる ECS インスタンスの作成

API は、従量課金 ECS インスタンスに加えて、サブスクリプションモードでの ECS インスタンス作成にも対応しています。サブスクリプションモードでの ECS インスタンス作成のプロセスは Alibaba Cloud のウェブサイトのものとは異なります。サブスクリプションモードで作成された ECS インスタンスの料金は、自動的に引き落とされます。ECS インスタンスを作成する前に、作成中に決済がスムーズに行われるよう、十分な口座残高やクレジット額があることを確認しておきます。

サブスクリプションモードの ECS インスタンスを作成する際、支払いオプションと期間を指定する必要があります。次のコードでは、期間は 1 ヶ月に設定されています。

```
request . set_Period ( 1 )  request . set_Instan  ceChargeTy  pe (‘
PrePaid ’)
```

サブスクリプションモードで ECS インスタンスを作成するためのコードは次のようになります。

```
#  create   one   prepay   ecs   instance .
 def   create_pre  pay_instan  ce ( image_id ,  instance_t  ype ,
 security_g  roup_id ):
     request  =  CreateInst  anceReques  t ();
     request . set_ImageI  d ( image_id )
     request . set_Securi  tyGroupId ( security_g  roup_id )
     request . set_Instan  ceType ( instance_t  ype )
     request . set_IoOpti  mized (' optimized ')
     request . set_System  DiskCatego  ry (' cloud_ssd ')
     request . set_Period ( 1 )
     request . set_Instan  ceChargeTy  pe (' PrePaid ')
     response  =  _send_requ  est ( request )
     instance_i  d  =  response . get (' InstanceId ')
     logging . info (" instance  % s   created   task   submit
 successful  ly .",  instance_i  d )
     return   instance_i  d ;
```

完全なコード

完全なコードは以下をご参照ください。リソースパラメーターの設定を使用します。

```
#  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   aliyunsdkc  ore   import   client
```

```
 from   aliyunsdke  cs . request . v20140526 .  CreateInst
 anceReques  t   import   CreateInst  anceReques  t
 from   aliyunsdke  cs . request . v20140526 .  DescribeIn
 stancesReq  uest   import   DescribeIn  stancesReq  uest
 from   aliyunsdke  cs . request . v20140526 .  StartInsta  nceRequest
   import   StartInsta  nceRequest
#  configurat  ion   the   log   output   formatter ,  if   you   want
   to   save   the   output   to   file ,
#  append  ", filename =' ecs_invoke . log '"  after   datefmt .
 gging . basicConfi  g ( 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   Secrect ', ' cn - beijing ')
  IMAGE_ID  = ' ubuntu1404  _64_40G_cl  oudinit_20  160727 . raw '
 INSTANCE_T  YPE  = ' ecs . s2 . large ' #  2c4g   generation   1
 SECURITY_G  ROUP_ID  = ' sg -****'
 INSTANCE_R  UNNING  = ' Running '
 def   create_ins  tance_acti  on ():
     instance_i  d  =  create_aft  er_pay_ins  tance ( image_id =
 IMAGE_ID ,  instance_t  ype = INSTANCE_T  YPE ,
                                          security_g  roup_id =
 SECURITY_G  ROUP_ID )
     check_inst  ance_runni  ng ( instance_i  d = instance_i  d )
 def   create_pre  pay_instan  ce_action ():
     instance_i  d  =  create_pre  pay_instan  ce ( image_id =
 IMAGE_ID ,  instance_t  ype = INSTANCE_T  YPE ,
                                        security_g  roup_id =
 SECURITY_G  ROUP_ID )
     check_inst  ance_runni  ng ( instance_i  d = instance_i  d )
#  create   one   after   pay   ecs   instance .
 def   create_aft  er_pay_ins  tance ( image_id ,  instance_t  ype ,
 security_g  roup_id ):
     request  =  CreateInst  anceReques  t ();
     request . set_ImageI  d ( image_id )
     request . set_Securi  tyGroupId ( security_g  roup_id )
     request . set_Instan  ceType ( instance_t  ype )
     request . set_IoOpti  mized (' optimized ')
     request . set_System  DiskCatego  ry (' cloud_ssd ')
     response  =  _send_requ  est ( request )
     instance_i  d  =  response . get (' InstanceId ')
     logging . info (" instance  % s   created   task   submit
 successful  ly .",  instance_i  d )
     return   instance_i  d ;
#  create   one   prepay   ecs   instance .
 def   create_pre  pay_instan  ce ( image_id ,  instance_t  ype ,
 security_g  roup_id ):
     request  =  CreateInst  anceReques  t ();
     request . set_ImageI  d ( image_id )
     request . set_Securi  tyGroupId ( security_g  roup_id )
     request . set_Instan  ceType ( instance_t  ype )
     request . set_IoOpti  mized (' optimized ')
     request . set_System  DiskCatego  ry (' cloud_ssd ')
     request . set_Period ( 1 )
     request . set_Instan  ceChargeTy  pe (' PrePaid ')
     response  =  _send_requ  est ( request
     instance_i  d  =  response . get (' InstanceId ')
     logging . info (" instance  % s   created   task   submit
 successful  ly .",  instance_i  d )
     return   instance_i  d ;
 def   check_inst  ance_runni  ng ( instance_i  d ):
     detail  =  get_instan  ce_detail_  by_id ( instance_i  d =
 instance_i  d ,  status = INSTANCE_R  UNNING )
```

```
    index = 0
    while detail is None and index & lt ; 60 :
        detail = get_instan ce_detail_ by_id ( instance_i d =
 instance_i d );
        time . sleep ( 10 )
    if detail and detail . get (' Status ') == ' Stopped ':
        logging . info (" instance % s is stopped now .")
        start_inst ance ( instance_i d = instance_i d )
        logging . info (" start instance % s job submit .")
    detail = get_instan ce_detail_ by_id ( instance_i d =
 instance_i d , status = INSTANCE_R UNNING )
    while detail is None and index & lt ; 60 :
        detail = get_instan ce_detail_ by_id ( instance_i d =
 instance_i d , status = INSTANCE_R UNNING );
        time . sleep ( 10 )
    logging . info (" instance % s is running now .",
 instance_i d )
    return instance_i d ;
 def start_inst ance ( instance_i d ):
    request = StartInsta nceRequest ()
    request . set_Instan ceId ( instance_i d )
    _send_requ est ( request )
#  output the instance owned in current region .
 def get_instan ce_detail_ by_id ( instance_i d , status ='
 Stopped '):
    logging . info (" Check instance % s status is % s ",
 instance_i d , status )
    request = DescribeIn stancesReq uest ()
    request . set_Instan ceIds ( json . dumps ([ instance_i d ]))
    response = _send_requ est ( request )
    instance_d etail = None
    if response is not None :
        instance_l ist = response . get (' Instances '). get ('
 Instance ')
        for item in instance_l ist :
            if item . get (' Status ') == status :
                instance_d etail = item
                break ;
        return instance_d etail ;
#  send open api request
 def _send_requ est ( request ):
    request . set_accept _format (' json ')
    try :
        response_s tr = clt . do_action ( request )
        logging . info ( response_s tr )
        response_d etail = json . loads ( response_s tr )
        return response_d etail
    except Exception as e :
        logging . error ( e )
 if __name__ == ' __main__ ':
    logging . info (" Create ECS by OpenApi !")
    create_ins tance_acti on ()
   #  create_pre pay_instan ce_action ()
```

4.3 インスタンスの管理

API を使用することで、リソースの作成や日常的な管理のための ECS コンソールの利用に加え、リソースの管理やカスタマイズもできます。 API は、より柔軟な ECS インスタンス管理と設定を提供します。 Alibaba Cloud では、API を一つの SDK 中にカプセル化することにより、

既存システムに ECS インスタンスの管理機能を統合します。 このトピックでは、Python の開発に基づき API を介してインスタンスを管理する方法について説明します。 Python の開発経験がない場合でも簡単に ECS インスタンスの開発ができます。

RAM ユーザーの "AccessKey" 取得

API を使用して ECS インスタンスを管理する場合は、"AccessKey" ("AccessKey ID" と " AccessKey Secret") が必要です。 クラウドサービスのセキュリティを維持するためには、RAM ユーザーを作成し、その "AccessKey" を生成し、ECS リソースのみを管理するために RAM ユーザーを認証する必要があります。 その後、RAM ユーザーとその "AccessKey" を使用して、API で ECS リソースを管理できるようになります。

RAM ユーザーの "AccessKey" を取得するには、次の手順に従います。

1. *RAM ユーザーを作成し、"AccessKey" を取得します。*
2. *RAM ユーザへの権限を直接付与します。* ECS リソースを管理するためには、RAM ユーザーに "AliyunECSFullAccess" を付与する必要があります。

ECS Python SDK のインストール

Python 実行環境がインストールされていることを確認します。 ここでは、Python 2.7 以上を使用しています。

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

権限がない場合は、” sudo ” に切り替えて続行します。

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

SDK のバージョンは 2.1.2 です。

Hello Alibaba Cloud

" hello_ecs_ api . py " ファイルを作成します。 SDK を使用するには、RAM ユーザーの "AccessKey" を使用して "AcsClient" オブジェクトをインスタンス化する必要があります。

注 :

"AccessKey" により、RAM ユーザーは Alibaba Cloud API にアクセスできるようになり、このユーザーのフルアクセスが許可されます。 "Access Key" は安全に保管します。

```
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 Secrect ', ' cn - beijing ')
```

"AcsClient" オブジェクトがインスタンス化されたら、最初のアプリケーションを開発できます。 アカウントが対応しているリージョンのリストを照会します。 詳しくは、「*DescribeRegions*」をご参照ください。

```
def hello_aliy un_regions ():
    request = DescribeRe gionsReque st ()
    response = _send_requ est ( request )
    region_lis t = response . get (' Regions '). get (' Region ')
    assert response is not None
    assert region_lis t is not None
    result = map ( _print_reg ion_id , region_lis t )
    logging . info (" region list : % s ", result )
def _print_reg ion_id ( item ):
    region_id = item . get (" RegionId ")
    return region_id
def _send_requ est ( request ):
    request . set_accept _format (' json ')
    try :
        response_s tr = clt . do_action ( request )
        logging . info ( response_s tr )
        response_d etail = json . loads ( response_s tr )
        return response_d etail
    except Exception as e :
        logging . error ( e )
hello_aliy un_regions ()
```

コマンドラインで、`" python hello_ecs_ api . py "` を実行して、対応しているリージョンのリストを取得します。 アウトプットは次のようになります。

```
[ 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 ']
```

現在のリージョンの ECS インスタンスリスト照会

インスタンスリストを照会する手順は、リージョンリストの場合と似ています。 必要な作業は、入力パラメーター "DescribeRegionsRequest" を `" DescribeIn stancesReq uest "` に置き換えるだけです。 クエリパラメーター一覧については「*DescribeInstances*」をご参照ください。

```
def list_insta nces () :
    request = DescribeIn stancesReq uest ()
    response = _send_requ est ( request )
    if response is not None :
        instance_l ist = response . get (' Instances '). get ('
Instance ')
        result = map ( _print_ins tance_id , instance_l ist )
        logging . info (" current region include instance % s
", result )
def _print_ins tance_id ( item ):
```

```
    instance_i  d  =  item . get (' InstanceId ');
    return   instance_i  d
```

アウトプットは以下の通りです。

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

API の全一覧については、「*ECS API の概要*」をご参照ください。 ディスク一覧の照会をしたい場合は、"DescribeInstancesRequest" を " DescribeDi sksRequest " に置き換えます。

完全なコード

ここで説明した操作の完全なコードを次に示します。

```
#  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   aliyunsdkc  ore   import   client
 from   aliyunsdke  cs . request . v20140526 .  DescribeIn
 stancesReq  uest   import   DescribeIn  stancesReq  uest
 from   aliyunsdke  cs . request . v20140526 .  DescribeRe
 gionsReque  st   import   DescribeRe  gionsReque  st
#  configurat  ion   the   log   output   formatter ,  if   you   want
   to   save   the   output   to   file ,
#  append  ", filename =' ecs_invoke . log '"  after   datefmt .
 logging . basicConfi  g ( 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   Secrect ', ' cn - beijing ')
#  sample   api   to   list   aliyun   open   api .
 def   hello_aliy  un_regions ():
     request  =  DescribeRe  gionsReque  st ()
     response  =  _send_requ  est ( request )
     if   response   is   not   None :
         region_lis  t  =  response . get (' Regions '). get (' Region
 ')
         assert   response   is   not   None
         assert   region_lis  t   is   not   None
         result  =  map ( _print_reg  ion_id ,  region_lis  t )
         logging . info (" region   list : % s ",  result )
#  output   the   instance   owned   in   current   region .
 def   list_insta  nces ():
     request  =  DescribeIn  stancesReq  uest ()
     response  =  _send_requ  est ( request )
     if   response   is   not   None :
         instance_l  ist  =  response . get (' Instances '). get ('
 Instance ')
         result  =  map ( _print_ins  tance_id ,  instance_l  ist )
```

```
            logging . info (" current   region   include   instance  % s
 ",  result )
 def   _print_ins  tance_id ( item ):
     instance_i  d  =  item . get (' InstanceId ');
     return   instance_i  d
 def   _print_reg  ion_id ( item ):
     region_id  =  item . get (" RegionId ")
     return   region_id
#  send   open   api   request
 def   _send_requ  est ( request ):
     request . set_accept  _format (' json ')
     try :
         response_s  tr  =  clt . do_action ( request )
         logging . info ( response_s  tr )
         response_d  etail  =  json . loads ( response_s  tr )
         return   response_d  etail
     except   Exception   as   e :
         logging . error ( e )
 if   __name__   == ' __main__ ':
     logging . info (" Hello   Aliyun   OpenApi !")
     hello_aliy  un_regions ()
     list_insta  nces ()
```

ECS での他の API 操作については、「*ECS API の操作*」をご参照ください。

4.4 インスタンスのリリース

ECS の重要な機能の一つがオンデマンドリソースの作成です。 サービスピークタイム中に要求に応じて弾性的にカスタムリソースを作成し、サービスコンピューティングが完了すればカスタムリソースをリリースすることができます。 ここでは、簡単に ECS インスタンスをリリースして弾力性を得る方法について説明します。

以下の API について触れています。

- *DeleteInstance*
- *ModifyInstanceAutoReleaseTime*
- *StopInstance*
- *DescribeInstances*

ECS インスタンスがリリースされると、インスタンスに使用された物理リソースがディスクやスナップショットを含め、再利用されます。 そのインスタンスのデータは完全に消失し、二度と復元できません。 データを保持しておきたい場合は、ECS インスタンスをリリースする前に、ディスクのスナップショットを作成しておくことを推奨します。 スナップショットは新規 ECS インスタンスの作成に直接使うことができます。

ECS インスタンスをリリースするためには、最初にインスタンスを停止しなければなりません。 ECS インスタンス停止後に他のアプリケーションが影響を受けた場合、インスタンスを再起動します。

ECS インスタンスの停止

ECS インスタンスを停止させるためには、インスタンスの課金方法に関係なく、"StopInstance " インターフェイスを使用します。 停止コマンドは以下のとおりです。 "ForceStop" パラメーターが "true" に設定されている場合は、ECS インスタンスがすぐに停止します。ただし、停電時と同様、データがディスクに必ずしも書き込まれるわけではありません。 したがって、インスタンスをリリースしたい場合は、"ForceStop" を "true" に設定します。

```
def   stop_insta  nce ( instance_i  d ,  force_stop = False ):
    '''
     stop   one   ecs   instance .
    : param   instance_i  d :  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  =  StopInstan  ceRequest ()
     request . set_Instan  ceId ( instance_i  d )
     request . set_ForceS  top ( force_stop )
     logging . info (" Stop  % s   command   submit   successful  ly
.",  instance_i  d )
     _send_requ  est ( request )
```

ECS インスタンスのリリース

ECS インスタンスのステータスが "停止" 以外のときにリリースした場合、エラーが発生します。

```
{" RequestId ":" 3C6DEAB4 - 7207 - 411F - 9A31 - 6ADE54C268  BE
 "," HostId ":" ecs - cn - hangzhou . aliyuncs . com "," Code ":"
 IncorrectI  nstanceSta  tus "," Message ":" The   current   status
 of   the   resource   does   not   support   this   operation ."}
```

ECS インスタンスのステータスが "停止" であるときに、インスタンスをリリースすることができます。 その API には、2 つのリクエストパラメーターがあります。

- InstanceId: インスタンス ID
- Force: このパラメーターが ”true” に設定されている場合、ステータスが "停止" になっていない場合でも、ECS インスタンスは強制的にリリースされます。 このパラメーターを設定するときは、ご注意ください。 誤ってリリースすると、サービスに影響を与える可能性があります。

ECS インスタンスをリリースするためのリクエストは、以下のとおりです。

```
def   release_in  stance ( instance_i  d ,  force = False ):
    '''
     delete   instance   according   instance   id ,  only   support
after   pay   instance .
    : param   instance_i  d :  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 = DeleteInst anceReques t ();
    request . set_Instan ceId ( instance_i d )
    request . set_Force ( force )
    _send_requ est ( request )
```

ECS インスタンスが正常にリリースされると、以下のレスポンスが返されます。

```
{" RequestId ":" 689E5813 - D150 - 4664 - AF6F - 2A27BB4986 A3 "}
```

ECS インスタンスの自動リリース時間の設定

ECS インスタンスの自動リリース時間を設定することで、インスタンス管理を簡単にすることができます。 設定時間に到達した時に、Alibaba Cloud はお客様の ECS インスタンスを自動的にリリースします。 ECS インスタンスの自動リリース時間を設定するためには ”ModifyInstanceAutoReleaseTime” を使用します。

 注 :

自動リリース時間は、ISO8601 標準規格の UTC 時間で記述します。 フォーマットは ”yyyy-MM-ddTHH:mm:ssZ” です。 秒 (ss) が 00 でない場合は、現在の分 (mm) から開始するように自動的に設定されます。 自動リリース時間は、現在時刻から少なくとも 30 分以上後であり、現在時刻から 3 年を超えないようにします。

```
def set_instan ce_auto_re lease_time ( instance_i d ,
time_to_re lease = None ):
   '''
    setting instance auto delete time
   : param instance_i d : instance id of the ecs
instance , like ' i -***'.
   : param time_to_re lease : 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 = ModifyInst anceAutoRe leaseTimeR equest ()
    request . set_Instan ceId ( instance_i d )
    if time_to_re lease is not None :
        request . set_AutoRe leaseTime ( time_to_re lease )
    _send_requ est ( request )
```

時間を設定するために `" set_instan ce_auto_re lease_time (' i - 1111 ', ' 2017 - 01 - 30T00 : 00 : 00Z ')"` コマンドを実行します。

その後、自動リリース時間の照会に "DescribeInstances" を使用できます。

```
def   describe_i  nstance_de  tail ( instance_i  d ):
    '''
     describe   instance   detail
    : param   instance_i  d :  instance   id   of   the   ecs
instance ,  like  ' i -***'.
    : return :
    '''
     request  =  DescribeIn  stancesReq  uest ()
     request . set_Instan  ceIds ( json . dumps ([ instance_i  d ]))
     response  =  _send_requ  est ( request )
     if   response   is   not   None :
         instance_l  ist  =  response . get (' Instances '). get ('
Instance ')
         if   len ( instance_l  ist ) >  0 :
             return   instance_l  ist [ 0 ]
def   check_auto  _release_t  ime_ready ( instance_i  d ):
    detail  =  describe_i  nstance_de  tail ( instance_i  d =
instance_i  d )
    if   detail   is   not   None :
        release_ti  me  =  detail . get (' AutoReleas  eTime ')
        return   release_ti  me
```

自動リリースのキャンセル

サービスの変更によって自動リリースをキャンセルしたい場合、自動リリース時間を "null" に変更するために以下のコマンドを実行します。

```
set_instan  ce_auto_re  lease_time (' i - 1111 ')
```

完全なサンプルコード

注:

ECS インスタンスをリリースするときは、注意して進めます。

```
#  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   aliyunsdkc  ore   import   client
 from   aliyunsdke  cs . request . v20140526 .  DeleteInst
 anceReques  t   import   DeleteInst  anceReques  t
 from   aliyunsdke  cs . request . v20140526 .  DescribeIn
 stancesReq  uest   import   DescribeIn  stancesReq  uest
 from   aliyunsdke  cs . request . v20140526 .  ModifyInst
 anceAutoRe  leaseTimeR  equest   import  \
     ModifyInst  anceAutoRe  leaseTimeR  equest
 from   aliyunsdke  cs . request . v20140526 .  StopInstan  ceRequest
   import   StopInstan  ceRequest
```

```
#  configurat  ion   the   log   output   formatter ,  if   you   want
   to   save   the   output   to   file ,
#  append  ", filename =' ecs_invoke . log '"  after   datefmt .
 logging . basicConfi  g ( 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   Secrect ', ' cn - beijing ')
 def   stop_insta  nce ( instance_i  d ,  force_stop = False ):
    '''
     stop   one   ecs   instance .
    : param   instance_i  d :  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  =  StopInstan  ceRequest ()
     request . set_Instan  ceId ( instance_i  d )
     request . set_ForceS  top ( force_stop )
     logging . info (" Stop  % s   command   submit   successful  ly
 .",  instance_i  d )
     _send_requ  est ( request )
 def   describe_i  nstance_de  tail ( instance_i  d ):
    '''
     describe   instance   detail
    : param   instance_i  d :  instance   id   of   the   ecs
 instance ,  like  ' i -***'.
    : return :
    '''
     request  =  DescribeIn  stancesReq  uest ()
     request . set_Instan  ceIds ( json . dumps ([ instance_i  d ]))
     response  =  _send_requ  est ( request )
     if   response   is   not   None :
         instance_l  ist  =  response . get (' Instances '). get ('
 Instance ')
         if   len ( instance_l  ist ) & gt ;  0 :
             return   instance_l  ist [ 0 ]
 def   check_auto  _release_t  ime_ready ( instance_i  d ):
     detail  =  describe_i  nstance_de  tail ( instance_i  d =
 instance_i  d )
     if   detail   is   not   None :
         release_ti  me  =  detail . get (' AutoReleas  eTime ')
         return   release_ti  me
 def   release_in  stance ( instance_i  d ,  force = False ):
    '''
     delete   instance   according   instance   id ,  only   support
 after   pay   instance .
    : param   instance_i  d :  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  =  DeleteInst  anceReques  t ();
     request . set_Instan  ceId ( instance_i  d )
     request . set_Force ( force )
     _send_requ  est ( request )
```

```
def set_instan ce_auto_re lease_time ( instance_i d ,
time_to_re lease = None ):
    '''
    setting instance auto delete time
    : param instance_i d : instance id of the ecs
instance , like ' i -***'.
    : param time_to_re lease : 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 = ModifyInst anceAutoRe leaseTimeR equest ()
    request . set_Instan ceId ( instance_i d )
    if time_to_re lease is not None :
        request . set_AutoRe leaseTime ( time_to_re lease )
    _send_requ est ( request )
    release_ti me = check_auto _release_t ime_ready (
instance_i d )
    logging . info (" Check instance % s auto release time
setting is % s . ", instance_i d , release_ti me )
def _send_requ est ( request ):
    '''
    send open api request
    : param request :
    : return :
    '''
    request . set_accept _format (' json ')
    try :
        response_s tr = clt . do_action ( request )
        logging . info ( response_s tr )
        response_d etail = json . loads ( response_s tr )
        return response_d etail
    except Exception as e :
        logging . error ( e )
if __name__ == ' __main__ ':
    logging . info (" Release ecs instance by Aliyun
OpenApi !")
    set_instan ce_auto_re lease_time (' i - 1111 ', ' 2017 - 01 -
28T06 : 00 : 00Z ')
    # set_instan ce_auto_re lease_time (' i - 1111 ')
    # stop_insta nce (' i - 1111 ')
    # release_in stance (' i - 1111 ')
    # release_in stance (' i - 1111 ', True )
```

ECS の他の API 操作については、「*ECS API の操作*」をご参照ください。

4.5 インスタンスの更新

サブスクリプション課金方法の ECS インスタンスにとって、ライフサイクルは重要です。

Alibaba Cloud は、ECS コンソールや ECS 購入ページに加え、有効期限の表示やインスタンス更新のAPIを提供します。

ここでは、以下のキー機能が関係しています。

・ 有効期限による ECS インスタンスの照会

- インスタンスの更新
- ECS インスタンスの自動更新時間の照会
- ECS インスタンスの自動更新時間の設定

ライフサイクルはサブスクリプションモードの ECS インスタンスにとって重要です。 ECS インスタンスの更新が間に合わず、そのインスタンスがロックもしくはリリースされてしまうと、お客様のサービスの継続性に影響するかもしれません。 API を使用して、リソースの有効期限を表示したりインスタンスを更新したりすることができます。

ここでは、以下の API について説明します。

- *DescribeInstances*
- *ModifyInstanceAutoRenewAttribute*

指定した期間内に有効期限が切れるインスタンスの照会

指定した期間内に有効期限が切れるインスタンスを照会するためには、"DescribeInstances" インターフェイスを使用して、フィルターパラメータである " ExpiredSta rtTime " と " ExpiredEnd Time " を設定します。 これらのパラメーターは、 ISO8601 規格の UTC 時間で、フォーマットは "yyyy-MM-ddTHH:mmZ" を用います。 システムは指定した期間内に有効期限が切れるインスタンスのリストを返します。 セキュリティグループでフィルタする場合は、セキュリティグループ ID を追加します。

```
INSTANCE_E XPIRED_STA RT_TIME_IN _UTC_STRIN G = ' 2017 - 01 -
22T00 : 00Z '
INSTANCE_E XPIRE_END_ TIME_IN_UT C_STRING = ' 2017 - 01 - 28T00 :
00Z '
def describe_n eed_renew_ instance ( page_size = 100 ,
page_numbe r = 1 , instance_i d = None ,
                                check_need _renew = True ,
security_g roup_id = None ):
    request = DescribeIn stancesReq uest ()
    if check_need _renew is True :
        request . set_Filter 3Key (" ExpiredSta rtTime ")
        request . set_Filter 3Value ( INSTANCE_E XPIRED_STA
RT_TIME_IN _UTC_STRIN G )
        request . set_Filter 4Key (" ExpiredEnd Time ")
        request . set_Filter 4Value ( INSTANCE_E XPIRE_END_
TIME_IN_UT C_STRING )
    if instance_i d is not None :
        request . set_Instan ceIds ( json . dumps ([ instance_i d
]))
    if security_g roup_id :
        request . set_Securi tyGroupId ( security_g roup_id )
    request . set_PageNu mber ( page_numbe r )
    request . set_PageSi ze ( page_size )
```

```
    return  _send_requ  est ( request )
```

ECS インスタンスの更新

サブスクリプションモードの ECS インスタンスのみの更新が可能です。 従量課金のインスタンスの更新はできません。 更新にはアカウント残高もしくはクレジットでのお支払いが必要です。 API の実行と同時に、料金の引き落としと請求が行われます。 ご自身のアカウントに十分な残高があることをご確認ください。

```
def  _renew_ins  tance_acti  on ( instance_i  d ,  period =' 1 '):
    request  =  RenewInsta  nceRequest ()
    request . set_Period ( period )
    request . set_Instan  ceId ( instance_i  d )
    response  =  _send_requ  est ( request )
    logging . info (' renew  % s  ready ,  output  is  % s  ',
instance_i  d ,  response )
```

インスタンスが更新された際に、料金は自動的に引き落とされます。 更新完了後、" InstanceId " に基づいてインスタンスの有効期限を照会することができます。 API が非同期的に実行されるため、有効期限の更新は 10 秒以内に行われます。

ECS インスタンスの自動更新の有効化

Alibaba Cloud は、期限切れリソースの管理費用の削減に寄与するために、サブスクリプションモードの ECS インスタンスで「自動更新機能」を提供しています。 自動更新の料金の引き落としは、有効期限より 9 日前の 08:00:00 に開始します。 初日に料金の引き落としに失敗した場合、引き落とし処理は料金が正常に引き落とされるまで毎日繰り返され、引き落としができずに 9 日後の期限を迎えた場合、リソースがロックされます。 アカウントに十分な残高とクレジットがあることをご確認ください。

- 自動更新設定の照会

  "OpenAPI" を使用して自動更新の照会と設定が行えます。 この API はサブスクリプションモードの ECS インスタンスにのみ対応しています。 従量課金インスタンスにこの API を使用した場合は、エラーが返されます。 サブスクリプション課金方法である ECS インスタンスの自動更新ステータスを、一度に最大 100 個照会することができます。 複数のインスタンス ID はコンマで区切ります。

  "DescribeInstanceAutoRenewAttribut" の入力パラメータはインスタンス ID です。

  InstanceId: サブスクリプションモードの ECS インスタンスを 100 個まで一括で照会することができます。 複数のインスタンス ID はコンマで区切ります。

```
#  check  the  instances  is  renew  or  not
 def  describe_a  uto_renew ( instance_i  ds ,  expected_a
 uto_renew = True ):
```

```
    describe_r  equest  =  DescribeIn  stanceAuto  RenewAttri
buteReques  t ()
    describe_r  equest . set_Instan  ceId ( instance_i  ds )
    response_d  etail  =  _send_requ  est ( request = describe_r
equest )
    failed_ins  tance_ids  = ''
    if   response_d  etail   is   not   None :
        attributes  =  response_d  etail . get (' InstanceRe
newAttribu  tes '). get (' InstanceRe  newAttribu  te ')
        if   attributes :
            for   item   in   attributes :
                auto_renew  _status  =  item . get (' AutoRenewE
nabled ')
                if   auto_renew  _status  ! =  expected_a
uto_renew :
                    failed_ins  tance_ids  +=  item . get ('
InstanceId ') + ','

describe_a  uto_renew (' i - 1111 , i - 2222 ')
```

以下の内容が返されます。

```
{" InstanceRe  newAttribu  tes ":{" InstanceRe  newAttribu  te ":[{"
 Duration ": 0 ," InstanceId ":" i - 1111 "," AutoRenewE  nabled ":
 false },{" Duration ": 0 ," InstanceId ":" i - 2222 "," AutoRenewE
 nabled ": false }]}," RequestId ":" 71FBB7A5 - C793 - 4A0D - B17E -
 D6B426EA74  6A "}
```

自動更新が設定されている場合、返される属性 " AutoRenewE nabled " は "true" です。

自動更新が設定されていない場合、属性は "false" になります。

- ECS インスタンスの自動更新の有効化とキャンセル

  ECS インスタンスの自動更新を有効にするには、3 つの入力パラメーターが必要です:

  - InstanceId: サブスクリプションモード中の ECS インスタンスを一括で 100 個まで照会することができます。 複数のインスタンス ID はコンマで区切ります。
  - Duration: 月単位で 1、2、3、6、または 12 を設定します。
  - AutoRenew: 自動更新を有効にするには "true" に設定します。 自動更新を無効にするには "false" に設定します。

```
def   setting_in  stance_aut  o_renew ( instance_i  ds ,
auto_renew  =  True ):
    logging . info (' execute   enable   auto   renew  ' +
instance_i  ds )
    request  =  ModifyInst  anceAutoRe  newAttribu  teRequest ();
    request . set_Durati  on ( 1 );
    request . set_AutoRe  new ( auto_renew );
    request . set_Instan  ceId ( instance_i  ds )
```

```
      _send_requ  est ( request )
```

操作が成功すると、以下のレスポンスが返されます:

```
{" RequestId ":" 7DAC9984 - AAB4 - 43EF - 8FC7 - 7D74C57BE4  6D "}
```

更新に成功すると、照会を実行することができます。 システムは更新時間と自動更新のステータス (TRUE / FALSE) を返します。

```
{" InstanceRe  newAttribu  tes ":{" InstanceRe  newAttribu  te ":[{"
 Duration ": 1 ," InstanceId ":" i - 1111 ," AutoRenewE  nabled ":
 true },{" Duration ": 1 ," InstanceId ":" i - 2222 ," AutoRenewE
 nabled ": true }]}," RequestId ":" 7F4D14B0 - D0D2 - 48C7 - B310 -
 B1DF713D43  31 "}
```

完全なサンプルコード

```
#  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   aliyunsdkc  ore   import   client
 from   aliyunsdke  cs . request . v20140526 .  DescribeIn
 stanceAuto  RenewAttri  buteReques  t   import  \
     DescribeIn  stanceAuto  RenewAttri  buteReques  t
 from   aliyunsdke  cs . request . v20140526 .  DescribeIn
 stancesReq  uest   import   DescribeIn  stancesReq  uest
 from   aliyunsdke  cs . request . v20140526 .  ModifyInst
 anceAutoRe  newAttribu  teRequest   import  \
     ModifyInst  anceAutoRe  newAttribu  teRequest
 from   aliyunsdke  cs . request . v20140526 .  RenewInsta  nceRequest
   import   RenewInsta  nceRequest
 logging . basicConfi  g ( 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   Secrect ', ' cn - beijing ')
#  data   format   in   UTC ,  only   support   passed   the   value
 for   minute ,  seconds   is   not   support .
 INSTANCE_E  XPIRED_STA  RT_TIME_IN  _UTC_STRIN  G  = ' 2017 - 01 -
 22T00 : 00Z '
 INSTANCE_E  XPIRE_END_  TIME_IN_UT  C_STRING  = ' 2017 - 01 - 28T00 :
 00Z '
 def   renew_job ( page_size = 100 ,  page_numbe  r = 1 ,  check_need
 _renew = True ,  security_g  roup_id = None ):
     response  =  describe_n  eed_renew_  instance ( page_size =
 page_size ,  page_numbe  r = page_numbe  r ,
                                          check_need  _renew =
 check_need  _renew ,
                                          security_g  roup_id =
 security_g  roup_id )
     response_l  ist  =  response . get (' Instances '). get ('
 Instance ')
```

```
    logging . info ("% s   instances   need   to   renew ",  str (
response . get (' TotalCount '))
    if   response_l  ist  >  0 :
        instance_i  ds  = ''
        for   item   in   response_l  ist :
            instance_i  d  =  item . get (' InstanceId ')
            instance_i  ds  +=  instance_i  d  + ','
            renew_inst  ance ( instance_i  d = instance_i  d )
        logging . info ("% s   execute   renew   action   ready ",
instance_i  ds )
def   describe_n  eed_renew_  instance ( page_size = 100 ,
page_numbe  r = 1 ,  instance_i  d = None ,
                                check_need  _renew = True ,
security_g  roup_id = None ):
    request  =  DescribeIn  stancesReq  uest ()
    if   check_need  _renew   is   True :
        request . set_Filter  3Key (" ExpiredSta  rtTime ")
        request . set_Filter  3Value ( INSTANCE_E  XPIRED_STA
RT_TIME_IN  _UTC_STRIN  G )
        request . set_Filter  4Key (" ExpiredEnd  Time ")
        request . set_Filter  4Value ( INSTANCE_E  XPIRE_END_
TIME_IN_UT  C_STRING )
    if   instance_i  d   is   not   None :
        request . set_Instan  ceIds ( json . dumps ([ instance_i  d
]))
    if   security_g  roup_id :
        request . set_Securi  tyGroupId ( security_g  roup_id )
    request . set_PageNu  mber ( page_numbe  r )
    request . set_PageSi  ze ( page_size )
    return   _send_requ  est ( request )
#  check   the   instances   is   renew   or   not
def   describe_i  nstance_au  to_renew_s  etting ( instance_i  ds ,
expected_a  uto_renew = True ):
    describe_r  equest  =  DescribeIn  stanceAuto  RenewAttri
buteReques  t ()
    describe_r  equest . set_Instan  ceId ( instance_i  ds )
    response_d  etail  =  _send_requ  est ( request = describe_r
equest )
    failed_ins  tance_ids  = ''
    if   response_d  etail   is   not   None :
        attributes  =  response_d  etail . get (' InstanceRe
newAttribu  tes '). get (' InstanceRe  newAttribu  te ')
        if   attributes :
            for   item   in   attributes :
                auto_renew  _status  =  item . get (' AutoRenewE
nabled ')
                if   auto_renew  _status  ! =  expected_a  uto_renew
:
                    failed_ins  tance_ids  +=  item . get ('
InstanceId ') + ','
    if   len ( failed_ins  tance_ids ) >  0 :
        logging . error (" instance  % s   auto   renew   not   match
  expect  % s .",  failed_ins  tance_ids ,
                      expected_a  uto_renew )
def   setting_in  stance_aut  o_renew ( instance_i  ds ,  auto_renew
= True ):
    logging . info (' execute   enable   auto   renew  ' +
instance_i  ds )
    request  =  ModifyInst  anceAutoRe  newAttribu  teRequest ();
    request . set_Durati  on ( 1 );
    request . set_AutoRe  new ( auto_renew );
    request . set_Instan  ceId ( instance_i  ds )
    _send_requ  est ( request )
```

```
    describe_i  nstance_au  to_renew_s  etting ( instance_i  ds ,
 auto_renew )
#  if  using  the  instance  id  can  be  found  means  the
 instance  is  not  renew  successful  ly .
 def  check_inst  ance_need_  renew ( instance_i  d ):
    response  =  describe_n  eed_renew_  instance ( instance_i  d =
 instance_i  d )
    if  response  is  not  None :
        return  response . get (' TotalCount ') ==  1
    return  False
#  Renew  an  instance  for  a  month
 def  renew_inst  ance ( instance_i  d ,  period =' 1 '):
    need_renew  =  check_inst  ance_need_  renew ( instance_i  d )
    if  need_renew :
        _renew_ins  tance_acti  on ( instance_i  d = instance_i  d ,
 period = period )
       #  describe_n  eed_renew_  instance ( instance_i  d =
 instance_i  d ,  check_need  _renew = False )
 def  _renew_ins  tance_acti  on ( instance_i  d ,  period =' 1 '):
    request  =  RenewInsta  nceRequest ()
    request . set_Period ( period )
    request . set_Instan  ceId ( instance_i  d )
    response  =  _send_requ  est ( request )
    logging . info (' renew  % s  ready ,  output  is  % s  ',
 instance_i  d ,  response )
 def  _send_requ  est ( request ):
    request . set_accept  _format (' json ')
    try :
        response_s  tr  =  clt . do_action ( request )
        logging . info ( response_s  tr )
        response_d  etail  =  json . loads ( response_s  tr )
        return  response_d  etail
    except  Exception  as  e :
        logging . error ( e )
 if  __name__  == ' __main__ ':
    logging . info (" Renew  ECS  Instance  by  OpenApi !")
   #  Query  whether  there  is  any  instance  that  needs
 to  be  renewed  within  the  specified  time  range .
    describe_n  eed_renew_  instance ()
   #  Renew  an  instance  by  direct  fee  deduction
    renew_inst  ance (' i - 1111 ')
   #  Query  the  status  of  automatic  renewal
   #  describe_i  nstance_au  to_renew_s  etting (' i - 1111 , i -
 2222 ')
   #  Set  automatic  instance  renewal
   #  setting_in  stance_aut  o_renew (' i - 1111 , i - 2222 ')
```

ECS の他の API 操作については、「*ECS API の操作*」をご参照ください。

4.6 API を使用したプリエンプティブルインスタンスの管理

ここは、Alibaba Cloud ECS SDK を使用して、プリエンプティブルインスタンスを迅速に作成し管理する方法について説明します。

準備

始める前に、以下をご確認ください。

・ どのインスタンスタイプとリージョンがビジネス要件を満たしているかを把握していること。

- Alibaba Cloud ECS SDK と呼び出し方法の基本的な知識を持っていること。詳細については、「SDK のドキュメント」をご参照ください。
- プリエンプティブルインスタンスに対応する ECS SDK のバージョンは 4.2.0 以降です。ここでは、"pom" の依存関係を変更する方法の例を示します。

```
< 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 >
```

リージョンと使用可能なインスタンスタイプの照会

DescribeZones インターフェイスを使用して、プリエンプティブルインスタンスを作成できるリージョンと使用可能なインスタンスタイプを照会します。サンプルコードは次のとおりです。

- OpenApiCaller.java

```
public class OpenApiCal ler {
  IClientPro file profile ;
  IAcsClient client ;
  public OpenApiCal ler () {
      profile = DefaultPro file . getProfile (" cn - hangzhou
",  AKSUtil . accessKeyI d ,  AKSUtil . accessKeyS ecret );
      client = new DefaultAcs Client ( profile );
  }
  public < T extends AcsRespons e > T doAction (
AcsRequest < T > var1 ) {
      try {
          return client . getAcsResp onse ( var1 );
      } catch ( Throwable e ) {
          e . printStack Trace ();
          return null ;
      }
  }
}
```

- DescribeZonesSample.java

```
public class DescribeZo nesSample {
  public static void main ( String [] args ) {
      OpenApiCal ler caller = new OpenApiCal ler ();
      DescribeZo nesRequest request = new DescribeZo
nesRequest ();
      request . setRegionI d (" cn - zhangjiako u ");// You can
  use DescribeRe gionsReque st to get the RegionId of
  each region
      request . setSpotStr ategy (" SpotWithPr iceLimit ");//
This field must be entered to query the availabili
ty of instance types
      request . setInstanc eChargeTyp e (" PostPaid ");// Post -
paid mode , preemptibl e instances must be post - paid
```

```
        DescribeZo  nesRespons  e   response  =  caller . doAction (
 request );
        System . out . println ( JSON . toJSONStri  ng ( response ));
  }
}
```

次の出力結果では、各リージョンで使用可能なインスタンスタイプ、ディスクタイプ、ネットワークタイプを確認できます。

```
{
  " requestId ": " 388D6321 - E587 - 470C - 8CFA - 8985E2963D  AE ",
  " zones ": [
     {
         " localName ": " China   North   3   Zone   A ",
         " zoneId ": " cn - zhangjiako  u - a ",
         " availableD  iskCategor  ies ": [
            " cloud_ssd ",
            " cloud_effi  ciency "
         ],
         " availableI  nstanceTyp  es ": [
            " 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 ",
            & quot ; ecs . se1 . xlarge & quot ;
        ],
        " availableR  esourceCre  ation ": [
            & quot ; VSWITCH & quot ;、
            " IoOptimize  d ",
            " Instance ",
            " Disk "
        ],
        " availableR  esources ": [
            {
                " dataDiskCa  tegories ": [
                    " cloud_ssd ",
                    " cloud_effi  ciency "
                ],
                " instanceGe  nerations ": [
                    " ecs - 3 ",
                    " ecs - 2 "
                ],
                " instanceTy  peFamilies ": [
                    " ecs . mn4 ",
                    " ecs . sn1 ",
                    & quot ; ecs . sn2 & quot ;、
                    " ecs . sn1ne ",
                    " ecs . xn4 ",
                    " ecs . i1 ",
                    " ecs . se1 ",
                    " ecs . e4 ",
                    " ecs . n4 ",
                    " ecs . se1ne ",
                    & quot ; ecs . sn2ne & quot ;
                ],
                " instanceTy  pes ": [
                    " ecs . n4 . 4xlarge ",
                    & quot ; ecs . sn2 . medium & quot ;、
                    " 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 ",
                    & quot ; ecs . se1 . xlarge & quot ;
                ],
                " ioOptimize  d ":  true ,
                " networkTyp  es ": [
                    " vpc "
                ],
                " systemDisk  Categories ": [
                    " cloud_ssd ",
                    " cloud_effi  ciency "
                ]
            }
        ],
        " availableV  olumeCateg  ories ": [
            " san_ssd ",
            " san_effici  ency "
        ]
    }
  ]
}
```

プリエンプティブルインスタンスの料金履歴の照会

DescripeSpotPriceHistory インターフェイスを使用して過去 30 日間のプリエンプティブルインスタンスの料金の変化を照会すると、最も費用対効果の高いリージョンとインスタンスタイプを見つけられます。 サンプルコード ("DescribeSpotPriceHistorySample.java") は次のとおりです。

```
public   class   DescribeSp  otPriceHis  torySample  {
    public   static   void   main ( String []  args ) {
        OpenApiCal  ler   caller  =  new   OpenApiCal  ler ();
        List < DescribeSp  otPriceHis  toryRespon  se . SpotPriceT
ype >  result  =  new   ArrayList < DescribeSp  otPriceHis
toryRespon  se . SpotPriceT  ype >();
        int   offset  =  0 ;
        while  ( true ) {
            DescribeSp  otPriceHis  toryReques  t   request  =  new
DescribeSp  otPriceHis  toryReques  t ();
            request . setRegionI  d (" cn - hangzhou ");// You   can
  use   DescribeRe  gionsReque  st   to   get   the   RegionId   of
each   region   where   preemptibl  e   instances   are   available
            request . setZoneId (" cn - hangzhou - b ");// You   must
  enter   the   zone
            request . setInstanc  eType (" ecs . sn2 . medium ");//
See   the   instance   types   returned   by   DescribeZo  nes ,
this   field   is   mandatory
            request . setNetwork  Type (" vpc ");// See   the
network   types   returned   by   DescribeZo  nes ,  this   field
is   mandatory
```

```
//             request . setIoOptim  ized (" optimized ");//
 Determines   if   the   instance   is   I / O   optimized ,  see
 IoOptimize  d   returned   by   DescribeZo  nes ,  this   field   is
   optional
//             request . setStartTi  me (" 2017 - 09 - 20T08 : 45 :
 08Z ");// The   price   start   time ,  optional ,  default   value
 :  within   3   days
//             request . setEndTime (" 2017 - 09 - 28T08 : 45 : 08Z
 ");// Price   end   time ,  optional
            request . setOffset ( offset );
            DescribeSp  otPriceHis  toryRespon  se   response  =
 caller . doAction ( request );
            if  ( response  ! =  null  &&  response . getSpotPri  ces
 () ! =  null ) {
                result . addAll ( response . getSpotPri  ces ());
            }
             if  ( response . getNextOff  set () ==  null  ||
 response . getNextOff  set () ==  0 ) {
                break ;
            }  else {
                offset  =  response . getNextOff  set ();
            }
        }
         if  (!  result . isEmpty ()) {
             for  ( DescribeSp  otPriceHis  toryRespon  se .
 SpotPriceT  ype   spotPriceT  ype  :  result ) {
                System . out . println ( spotPriceT  ype .
 getTimesta  mp () + "---> spotPrice :" +  spotPriceT  ype .
 getSpotPri  ce () + "----> originPric  e :" +  spotPriceT  ype .
 getOriginP  rice ());
            }
             System . out . println  ( result . size  () ) 。
        }  else {
        }
    }
}
```

返された結果のサンプルは以下です。

```
 2017 - 09 - 26T06 : 28 : 55Z  --->  spotPrice : 0 . 24  ---->
 originPric  e : 1 . 2
 2017 - 09 - 26T14 : 00 : 00Z  --->  spotPrice : 0 . 36  ---->
 originPric  e : 1 . 2
 2017 - 09 - 26T15 : 00 : 00Z  --->  spotPrice : 0 . 24  ---->
 originPric  e : 1 . 2
 2017 - 09 - 27T14 : 00 : 00Z  --->  spotPrice : 0 . 36  ---->
 originPric  e : 1 . 2
 2017 - 09 - 27T15 : 00 : 00Z  --->  spotPrice : 0 . 24  ---->
 originPric  e : 1 . 2
 2017 - 09 - 28T14 : 00 : 00Z  --->  spotPrice : 0 . 36  ---->
 originPric  e : 1 . 2
 2017 - 09 - 28T15 : 00 : 00Z  --->  spotPrice : 0 . 24  ---->
 originPric  e : 1 . 2
 2017 - 09 - 29T06 : 28 : 55Z  --->  spotPrice : 0 . 24  ---->
 originPric  e : 1 . 2
```

このプロセスを繰り返して、各ゾーンのインスタンスタイプの料金傾向と最近の料金を確認します。

注：

平均価格や最高価格を用いて、このプリエンプティブルインスタンスをまかなえるかどうかを判断できます。 また、より合理的なデータモデルを使用して、料金データの履歴を分析し、インスタンスのタイプとゾーンを自由に調整して、費用対効果を最大化することができます。

プリエンプティブルインスタンスの作成

プリエンプティブルインスタンスを作成する前に、次の作業を完了する必要があります。

- カスタムイメージを使用してプリエンプティブルインスタンスを作成するには、すでに *CreateImage* してある必要があります。
- コンソールで、「セキュリティグループの作成」を実行するか、*CreateSecurityGroup* を使用してセキュリティグループを作成します。 次に、セキュリティグループの ID (SecurityGroupId) を取得します。
- コンソールで 「*VPC と VSwitch を作成する*」か、*CreateVpc* と *CreateVSwitch* インターフェイスを使用します。 次に、"VSwitchID (VSwitchId)" を取得します。

プリエンプティブルインスタンスを作成するには、*CreateInstance* を使用します。 サンプルコード ("CreateInstaneSample.java") は次のとおりです。

```
public   class   CreateInst  aneSample  {
    public   static   void   main ( String []  args ) {
        OpenApiCal  ler   caller  =  new   OpenApiCal  ler ();
        CreateInst  anceReques  t   request  =  new   CreateInst
anceReques  t ();
        request . setRegionI  d (" cn - hangzhou ");// The   region
ID
        request . setZoneId (" cn - hangzhou - b "); // The   zone
ID
        request . setSecurit  yGroupId (" sg - bp11nhf94i  vkdxwb2gd4
");// The   ID   of   the   security   group
        request . setImageId (" centos_7_0  3_64_20G_a  libase_201
70818 . vhd ");// We   recommend   that   you   select   a   custom
image   you   have   prepared   in   this   region
        request . setVSwitch  Id (" vsw - bp164cyont  hfudn9kj5b  r
");// For   VPC ,  the   VSwitch   ID   is   required
        request . setInstanc  eType (" ecs . sn2 . medium "); //
Enter   the   instance   type   you   want   to   purchase
        request . setIoOptim  ized (" optimized ");// See   the
parameters   returned   by   DescirbeZo  nes
        request . setSystemD  iskCategor  y (" cloud_ssd ");// See
the   parameters   returned   by   DescirbeZo  nes ,  select   one :
cloud_ssd ,  cloud_effi  ciency ,  or   cloud
        request . setSystemD  iskSize ( 40 );
        request . setInstanc  eChargeTyp  e (" PostPaid ");// Post -
paid   is   required   for   preemptibl  e   instances
        request . setSpotStr  ategy (" SpotWithPr  iceLimit ");//
SpotWithPr  iceLimit :  bid   mode ,  SpotAsPric  eGo :  no   bids ,
the   maximum   Pay - As - You - Go   price
        request . setSpotPri  ceLimit ( 0 . 25F );// This   applies
only   to   SpotWithPr  iceLimit .  Set   the   maximum   price   you
  are   willing   to   pay ,  units :  USD / hour .  An   instance
```

```
    is   created   when   this   price   is   higher   than   the
 current   market   price
        CreateInst  anceRespon  se   response  =  caller . doAction (
 request );
        System . out . println  ( response . getInstanc  eId  () ) 。
    }
}
```

プリエンプティブルインスタンスの復元

料金の変化や需要と供給の変化により、プリエンプティブルインスタンスを強制的に復元させることができます。 そのような場合、プリエンプティブルインスタンスの操作は中断されます。 リリースされる前に、プリエンプティブルインスタンスのステータスはロック中になり、インスタンスが自動的に復元されることを知らせるプロンプトが表示されます。 インスタンスが自動的に復旧ステータスに処理されるよう、終了ロジックを設計することができます。

ここでは、次の方法を使用して、プリエンプティブルインスタンスの中断ステータスとロックステータスを決定することができます。

- 「インスタンスのメタデータ」からこの情報を取得します。 次のコマンドを実行します。

  ```
  curl  ' http :// 100 . 100 . 100 . 200 / latest / meta - data /
  instance / spot / terminatio  n - time '
  ```

  返された結果が空白の場合は、インスタンスを引き続き使用できることを示しています。 結果が返されない場合は、インスタンスを引き続き使用できることを示しています。 返された結果に "YYYY-MM-DDTHH:mm:ssZ" 形式で UTC タイムスタンプ情報が含まれている場合、(例: `" 2015 - 01 - 05T18 : 02 : 00Z `")、指定した時間にインスタンスがリリースされることを示しています。
- *DescribeInstances*によって返された " OperationL ocks " 情報を使用して、インスタンスが " Awaiting Recovery " ステータスにあるかどうかを確認できます。 サンプルコード ("DescribeInstancesSample.java") は次のとおりです。

  ```
  public  class  DescribeIn  stancesSam  ple  {
    public  static  void  main ( String []  args )  throws
  Interrupte  dException  {
        OpenApiCal  ler  caller  =  new  OpenApiCal  ler ();
        JSONArray  allInstanc  es  =  new  JSONArray ();
        allInstanc  es . addAll ( Arrays . asList (" i - bp18hgfai8
  ekoqwo0y2n ", " i - bp1ecbyds2  4ij63w146c "));
        while  (!  allInstanc  es . isEmpty ()) {
            DescribeIn  stancesReq  uest  request  =  new
  DescribeIn  stancesReq  uest ();
            request . setRegionI  d (" cn - hangzhou ");
            request . setInstanc  eIds ( allInstanc  es . toJSONStri
  ng ());// Specify  the  instance  ID ,  maximum  efficiency
            DescribeIn  stancesRes  ponse  response  =  caller .
  doAction ( request );
            List < DescribeIn  stancesRes  ponse . Instance >
  instanceLi  st  =  response . getInstanc  es ();
  ```

```
            if ( instanceLi st != null && ! instanceLi st .
isEmpty ()) {
                for ( DescribeIn stancesRes ponse . Instance
instance : instanceLi st ) {
                    System . out . println (" result : instance :" +
instance . getInstanc eId () + ", az :" + instance . getZoneId
());
                    if ( instance . getOperati onLocks () != null
) {
                        for ( DescribeIn stancesRes ponse .
Instance . LockReason lockReason : instance . getOperati
onLocks ()) {
                            System . out . println (" instance :" +
instance . getInstanc eId () + "--> lockReason :" + lockReason .
getLockRea son () + ", vmStatus :" + instance . getStatus ());
                            if (" Recycling ". equals ( lockReason .
getLockRea son ())) {
                                // do your action
                                System . out . println (" spot
instance will be recycled immediatel y , instance id :"
+ instance . getInstanc eId ());
                                allInstanc es . remove ( instance .
getInstanc eId ());
                            }
                        }
                    }
                }
                System . out . print (" try describeIn stances
again later ...") ;
                Thread . sleep ( 2 * 60 * 1000 );
            } else {
                break ;
            }
        }
    }
}
```

復旧がトリガーされたときの出力結果は、次のとおりです。

```
instance : i - bp1ecbyds2 4ij63w146c --> lockReason : Recycling ,
vmStatus : Stopped
spot instance will be recycled immediatel y , instance
id : i - bp1ecbyds2 4ij63w146c
```

その他の操作

プリエンプティブルインスタンスを開始、停止、リリースすることができます。 これらの操作は従量課金インスタンスと同じです。 詳細については、以下をご参照ください。

- インスタンスの開始 *StartInstance*
- インスタンスの停止 *StopInstance*
- インスタンスのリリース *DeleteInstance*