

Alibaba Cloud Auto Scaling

Best practices

Issue: 20190417

Legal disclaimer

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

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

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

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

Generic conventions

Table -1: Style conventions

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

Style	Description	Example
<code>{}</code> or <code>{a b}</code>	It indicates that it is a required value, and only one item can be selected.	<code>switch {stand slave}</code>

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 Reduce costs by configuring scaling policies.....	1
2 Automatically add ECS instances of a scaling group to a Redis instance whitelist.....	5

1 Reduce costs by configuring scaling policies

This topic describes how to configure policies to reduce application costs by using multiple ECS instance types and multiple available zones. These policies increase the success rate of automatic scaling and also reduce costs.

Background information

Auto Scaling supports multiple instance types and allows you to specify alternative ECS instance types in a scaling configuration. Auto Scaling will attempt to create the ECS instance type that has the highest priority. If the ECS instance type of the highest priority is unavailable, Auto Scaling will attempt to create an ECS instance type of the next highest priority, until the specified number of ECS instances are created. If you configure multiple ECS instance types, Auto Scaling can still create instances when certain instance types are unavailable. This ensures that scaling activities can still be executed. During peak traffic periods, you may need to quickly create ECS instances to bear traffic. In this case, performance is prioritized over creating instances that comply with your cost requirements. You can set multiple instance types so that Auto Scaling can scale out to meet urgent demands.

Auto Scaling supports multiple zones. You can specify multiple VSwitches when creating a scaling group. If the zone where a VSwitch is located does not have sufficient ECS instances in stock, Auto Scaling will create ECS instances in other zones to ensure that instances are sufficient to execute scaling activities. After configuring multiple zones, you can also configure scaling policies based on your actual business needs. Multi-zone scaling policies include priority policies, distribution balancing policies, and cost optimization policies. For more information, see [Multi-zone scaling policies](#).

As [preemptible instances](#) are subject to market prices, failures of bidding such instances may delay scale-out activities, affecting your businesses. To avoid negative impact on your businesses, you can choose to configure a cost optimization policy. When Auto Scaling fails to create preemptible instances, the scaling group will automatically create the same number of Pay-As-You-Go instances of the same type as the preemptible instances. This ensures a high success rate for scaling activities and controls costs. You can configure a cost optimization policy and multiple instances types at the same time to increase the scaling success rate. Scaling groups that apply

cost optimization policies will attempt to create ECS instances based on the unit price of vCPUs, from low to high. Even if you have not specified any preemptible instances, this method ensures that you can use ECS instance resources at the lowest cost.

Notes

- Multi-zone scaling policies are available only for scaling groups whose network type is set to VPC.
- The multi-zone scaling policy of a scaling group cannot be modified.

Preparations

Before creating a scaling group that applies a cost optimization policy, make sure you have completed the following preparations:

- You have created a VPC. For more information, see [Create a VPC](#).
- You have created multiple VSwitches in different zones of the created VPC. For more information, see [Create VSwitches](#).

Procedure



Note:

The following section introduces multi-zone scaling policies. For more information about creating a scaling group, see [Use custom scaling configurations to create scaling groups](#) or [Use launch templates to create scaling groups](#).

1. Log on to the [Auto Scaling console](#).
2. On the Create Scaling Group page, set Network Type to VPC and select multiple VSwitches belonging to a VPC.



Note:

Because a VSwitch belongs to only one zone, you can create ECS instances in multiple zones after selecting multiple VSwitches. This helps you make use of ECS instance inventory in different zones.

3. On the Create Scaling Group page, set Multi-Zone Scaling Policy to Cost Optimization.
4. Configure other parameters as required.

5. [Create a scaling configuration](#) for the scaling group. In the configuration, set Billing Method to Preemptible Instance and choose multiple instance types (no more than 10 types).

**Note:**

- We recommend that you choose multiple similar instance types, taking into consideration factors such as vCPU, memory, processor frequency, internal network bandwidth, and packet transmission rate of the internal network.
- We recommend that you set an upper limit to the price. If you use automatic bidding, the scaling group will bid for and create preemptible instances at the market price.
- The specifications of I/O optimized instances and non-I/O optimized instances vary greatly from each other. Even if you choose these two types of instances at the same time, the scaling success rate cannot be increased significantly.

6. Configure other parameters as required.
7. Enable the created scaling group.
8. [Create a scaling rule](#) that adds an ECS instance to the scaling group.

Cost control result

Assume that in the preceding [procedure](#), you have specified VSwitches in Qingdao Zone B and Qingdao Zone C for the scaling group, and specified instance types ecs.sn1.large and ecs.sn1.xlarge for the scaling configuration. As Billing Method was set to Preemptible Instance, each instance type has two prices: the vCPU unit price for preemptible instances and the vCPU unit price for Pay-As-You-Go instances.

**Notice:**

The prices listed in this topic are only used as examples. Refer to the instance purchase page for actual prices.

After the instance types and billing method are set, you now have four instance creation plans that are sorted by vCPU unit price from low to high. The following table lists the plans.

No.	Instance type	Billing method	vCPU	Market price	vCPU unit price
Plan 1	ecs.sn1.xlarge	Preemptible instance	8	0.158/hour	0.01975/hour
Plan 2	ecs.sn1.large	Preemptible instance	4	0.088/hour	0.022/hour
Plan 3	ecs.sn1.xlarge	Pay-As-You-Go	8	1.393/hour	0.174125/hour
Plan 4	ecs.sn1.large	Pay-As-You-Go	4	0.697/hour	0.17425/hour

Expected instance creation process: When a scaling activity starts, the scaling group first attempts to create instances based on plan 1. If the group fails to create instances in Zone B and Zone C, it attempts to create instances based on plan 2, plan 3, and plan 4 in sequence until instance creation succeeds.

[Execute a scaling rule](#) to trigger the scaling activity of adding one ECS instance to the scaling group. Then, log on to the Auto Scaling console and navigate to the ECS Instances page of the scaling group. Click the ECS instance that has been created to view details. The billing method is Preemptible Instance and the instance type is ecs.sn1.xlarge, which indicates reduced costs.

2 Automatically add ECS instances of a scaling group to a Redis instance whitelist

This topic describes how to add scaled-out ECS instances to the whitelist of an ApsaraDB for Redis (Redis) instance by using Auto Scaling, Message Service (MNS), and Function Compute. You can refer to this topic to learn how to perform more automated operations.

Background information

Auto Scaling is integrated with Server Load Balancer (SLB) and Relational Database Service (RDS), but is not integrated with ApsaraDB for Redis. If your business data is stored in a Redis instance, you need to perform certain operations to allow ECS instances of a scaling group to access the Redis instance. Manually adding every created ECS instance to the Redis instance whitelist is time-consuming and prone to mistakes. When there are a large number of instances, manually adding these instances incurs large costs.

In this case, you can create a lifecycle hook in a scaling group. During a scale-out process, the lifecycle hook automatically sends messages to a specific MNS topic. Then, an MNS topic trigger in Function Compute executes the uploaded code to add the scaled-out ECS instances to the whitelist of the Redis instance.

Preparations

Before you start, ensure that you have activated [ApsaraDB for Redis](#), [Message Service](#), [Auto Scaling](#), and [Function Compute](#).

Notes

- We recommend that you create MNS topics, MNS queues, Function Compute services, Function Compute functions, and Redis instances in the same region.
- This topic provides sample code written in Java. You can follow these best practices with other programming languages.

1. Procedure in ApsaraDB for Redis

1. Log on to the [ApsaraDB for Redis console](#).
2. [Create a Redis instance](#) that serves as the database for automatically created ECS instances.

3. View the Redis instance whitelist to check the whitelist status before code execution.

2. Procedure in Message Service

1. Log on to the [Message Service console](#).
2. [Create an MNS topic](#) that will be used as a function trigger. Set the topic name to `fc - trigger`.
3. [Create an MNS queue](#) that will be used to receive function execution results. Set the queue name to `fc - callback`. In the [sample code](#), `QUEUE_NAME` is set to this queue name to send function execution results.

3. Procedure in Auto Scaling

1. Log on to the [Auto Scaling console](#).
2. Create a scaling group. For more information, see [Use custom scaling configurations to create scaling groups](#) or [Use launch templates to create scaling groups](#).
3. [Create a lifecycle hook](#).
 - a. Set Scaling Activity Type to Scale-Out to notify of scale-out activities.
 - b. Set Notification Method to MNS Topic. Compared with the MNS Queue method, the MNS Topic method can be used to notify multiple subscribers and execute various operations.
 - c. Set MNS Topic to `fc - trigger`. When automatically created ECS instances enter the Adding: Wait state, this topic is used to execute code to add the instances to the whitelist of a Redis instance.
 - d. Configure other parameters as required.

4. Procedure in Function Compute

1. Log on to the [Function Compute console](#).
2. [Create a service](#) for functions to be executed. Set the service name to `as - hook - mns - fc - redis`.

3. Under the created service, create a function to subscribe to the MNS topic, and upload the code. For more information, see [Create a function](#).
 - a. On the Function Template page, select Empty Function.
 - b. On the Trigger Configuration page, select MNS Topic Trigger from the Trigger Type drop-down list and configure other parameters as required.
 - c. On the Configure Function Settings page, set Service Name to `as - hook - mns - fc - redis` and Function Handler to `fc . Example :: handleRequest`, and configure other parameters as required.

**Note:**

- Function Handler varies with code. Set this parameter based on the actual conditions.
- In the examples provided in this topic, a JAR package is uploaded to add automatically created ECS instances to the whitelist of a Redis instance. For more information about Java programming descriptions, see [Java programming in Function Compute](#).

- d. On the Permission Configuration page, grant permissions on necessary resources to the function and allow Message Service to call functions.

**Note:**

We recommend that you follow the principle of least privilege and only grant necessary permissions to prevent potential risks.

- e. On the Preview page, confirm Function Information and Trigger Information, and click Create.

Expected results

After the configurations are completed, the expected results are as follows:

1. When the scaling conditions are met, the scaling group triggers the scaling activity and automatically creates ECS instances.
2. The lifecycle hook suspends the scaling activity and sends a message to the MNS topic.
3. In Function Compute, the MNS topic trigger triggers the function, fills in the Java code with the message content (such as ECS IDs), and executes the code.

4. When the code is being executed, the private IP addresses of the ECS instances are obtained and then added to the default group of the Redis instance whitelist.
5. The result of the executed code is sent to MNS queue `fc - callback`. You can view the result details in the [Message Service console](#). If `success` in the message content is `true` as shown in the following figure, the ECS instances have been added to the whitelist of the Redis instance.

You can also continue to consume messages in the MNS queue, such as obtaining the `success`, `LifecycleHookId`, and `LifecycleActionToken` parameters.

With these parameters, you can stop a lifecycle hook in advance.

The preceding best practices are for your reference. You can also use other necessary services to manage resources in your scaling groups with more flexibility.

Sample code

The sample code is for reference only. Use and test the code based on your actual conditions. Maven is used to manage the Java files that are used to add the instances to the whitelist. The directory structure is as follows:

Maven dependencies:

```
<? xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.aliyun.fc.wujin</groupId>
  <artifactId>demo</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>com.aliyun</groupId>
      <artifactId>aliyun-java-sdk-ecs</artifactId>
      <version>4.10.1</version>
    </dependency>
    <dependency>
      <groupId>com.aliyun.fc.runtime</groupId>
      <artifactId>fc-java-core</artifactId>
      <version>1.0.0</version>
    </dependency>
    <dependency>
      <groupId>com.aliyun</groupId>
      <artifactId>aliyun-java-sdk-core</artifactId>
      <version>3.2.6</version>
    </dependency>
  </dependencies>
```

```

    < dependency >
      < groupId > com . aliyun </ groupId >
      < artifactId > aliyun - java - sdk - r - kvstore </
artifactId >
      < version > 2 . 0 . 3 </ version >
    </ dependency >
    < dependency >
      < groupId > com . alibaba </ groupId >
      < artifactId > fastjson </ artifactId >
      < version > 1 . 2 . 25 </ version >
    </ dependency >
    < dependency >
      < groupId > org . springfram ework </ groupId >
      < artifactId > spring - context </ artifactId >
      < version > 4 . 2 . 5 . RELEASE </ version >
    </ dependency >
    < dependency >
      < groupId > org . apache . httpcompon ents </ groupId >
      < artifactId > httpclient </ artifactId >
      < version > 4 . 5 . 2 </ version >
    </ dependency >
    < dependency >
      < groupId > org . apache . commons </ groupId >
      < artifactId > com . springsour ce . org . apache .
commons . lang </ artifactId >
      < version > 2 . 6 . 0 </ version >
    </ dependency >
    < dependency >
      < groupId > com . aliyun . mns </ groupId >
      < artifactId > aliyun - sdk - mns </ artifactId >
      < version > 1 . 1 . 8 . 4 </ version >
    </ dependency >
  </ dependenci es >

  < build >
    < plugins >
      < plugin >
        < artifactId > maven - assembly - plugin </ artifactId
>
        < version > 3 . 1 . 0 </ version >
        < configurat ion >
          < descriptor Refs >
            < descriptor Ref > jar - with - dependenci
es </ descriptor Ref >
            </ descriptor Refs >
            < appendAsse mblyId > false </ appendAsse mblyId
>
            <!-- this is used for not append id to the jar
name -->
          </ configurat ion >
          < executions >
            < execution >
              < id > make - assembly </ id > <!-- this
is used for inheritanc e merges -->
              < phase > package </ phase > <!-- bind to
the packaging phase -->
              < goals >
                < goal > single </ goal >
              </ goals >
            </ execution >
          </ executions >
        </ plugin >
        < plugin >
          < groupId > org . apache . maven . plugins </ groupId
>

```

```

        < artifactId > maven - compiler - plugin </ artifactId >
    >
        < configurat ion >
            < source > 1 . 8 </ source >
            < target > 1 . 8 </ target >
        </ configurat ion >
    </ plugin >
</ plugins >
</ build >

</ project >

```

Example of *Example . java* :

```

package    fc ;

import    com . alibaba . fastjson . JSON ;
import    com . alibaba . fastjson . TypeRefere nce ;
import    com . aliyun . fc . runtime . Context ;
import    com . aliyun . fc . runtime . StreamRequ estHandler ;
import    com . aliyun . mns . client . CloudAccou nt ;
import    com . aliyun . mns . client . CloudQueue ;
import    com . aliyun . mns . client . MNSClient ;
import    com . aliyun . mns . model . Message ;
import    com . aliyuncs . DefaultAcs Client ;
import    com . aliyuncs . IAcsClient ;
import    com . aliyuncs . ecs . model . v20140526 . DescribeIn
stancesReq uest ;
import    com . aliyuncs . ecs . model . v20140526 . DescribeIn
stancesRes ponse ;
import    com . aliyuncs . exceptions . ClientExce ption ;
import    com . aliyuncs . profile . DefaultPro file ;
import    com . aliyuncs . profile . IClientPro file ;
import    com . aliyuncs . r_kvstore . model . v20150101 .
DescribeSe curityIpsR equest ;
import    com . aliyuncs . r_kvstore . model . v20150101 .
DescribeSe curityIpsR esponse ;
import    com . aliyuncs . r_kvstore . model . v20150101 .
ModifySecu rityIpsReq uest ;
import    model . FCResult ;
import    model . HookModel ;
import    model . MnsMessage Model ;
import    org . apache . commons . codec . binary . Base64 ;
import    org . apache . commons . lang . StringUtil s ;
import    org . springfram ework . util . Collection Utils ;

import    java . io . IOExceptio n ;
import    java . io . InputStrea m ;
import    java . io . OutputStre am ;
import    java . util . ArrayList ;
import    java . util . Arrays ;
import    java . util . List ;
public    class    Example implements    StreamRequ estHandler {

    /**
     * The VPC network . Do not change the value .
     */
    private    static    final    String    VPC_NETWOR K
    = " vpc ";

    private    static    final    String    CHAR_SET
    = " UTF - 8 ";

```

```

/**
 * The size of the array receiving input
information . A value of 4096 is sufficient in most
cases .
 */
private static final Integer MAX_BYTE_LENGTH
= 4096 ;

/**
 * The default group of the Redis instance
whitelist .
 */
private static final String DEFAULT_SECURITY_GROUP_NAME
= " default ";

/**
 * The method for modifying the REDIS instance
whitelist .
 */
private static final String MODIFY_MODE_APPEND
= " Append ";

/**
 * The address that the MNS client uses to send
messages .
 */
private static final String MNS_ENDPOINT
= " http ://% s . mns .% s . aliyuncs . com /";

/**
 * The Redis instance ID to be added . Set the
value based on the actual conditions .
 */
private static final String REDIS_ID
= "";

/**
 * The name of the queue that receives the result
of the executed function . Set the value based on
the actual conditions .
 */
private static final String QUEUE_NAME
= " wujin - fc - callback ";

/**
 * The Alibaba Cloud account ID . Set the value
based on the actual conditions .
 */
private static final Long USER_ID
= 1111111111 1111111111L ;

/**
 * The region to which the scaling group , Message
Service , and Function Compute belong . Set the value
based on the actual conditions .
 */
private static final String REGION_ID
= " cn - hangzhou ";

@Override
public void handleRequest ( InputStream inputStream
, OutputStream outputStream , Context context ) {
    FCResult result = new FCResult ();

```

```

        String akId = context . getExecuti onCredenti als ().
getAccessK eyId ();
        String akSecret = context . getExecuti onCredenti als
(). getAccessK eySecret ();
        String securityTo ken = context . getExecuti
onCredenti als (). getSecurit yToken ();
        try {
            // Obtain the content that an MNS trigger
used to trigger function execution
            String input = readInput ( inputStrea m );
            MnsMessage Model mnsMessage Model = JSON .
parseObjec t ( input ,
                new TypeRefere nce < MnsMessage Model >() {
                });
            if ( mnsMessage Model == null ) {
                result . setSuccess ( false );
                result . setMessage ( " mnsMessage Model is null
");
                sendMns ( akId , akSecret , securityTo ken ,
result . toString ());
                return ;
            }
            HookModel contentMod el = mnsMessage Model .
getContent ();
            if ( contentMod el == null ) {
                result . setSuccess ( false );
                result . setMessage ( " contentMod el is null " );
                sendMns ( akId , akSecret , securityTo ken ,
result . toString ());
                return ;
            }
            IAcsClient client = buildClie nt ( akId , akSecret
, securityTo ken );
            // Obtain the private IP addresses of ECS
instances that are being scaled .
            List < String > privateIps = getInstanc esPrivateI
ps ( contentMod el . getInstanc eIds (), client );
            if ( Collection Utils . isEmpty ( privateIps )) {
                result . setSuccess ( false );
                result . setMessage ( " privateIps is empty " );
                sendMns ( akId , akSecret , securityTo ken ,
result . toString ());
                return ;
            }
            List < String > needAppend Ips = filterPriv
ateIpsForA ppend ( privateIps , client );
            if ( ! Collection Utils . isEmpty ( needAppend Ips ))
            {
                modifySecu rityIps ( client , needAppend Ips );
                result . setLifecyc leHookId ( contentMod el .
getLifecyc leHookId ());
                result . setLifecyc leActionTo ken ( contentMod el
. getLifecyc leActionTo ken ());
                sendMns ( akId , akSecret , securityTo ken ,
result . toString ());
            }
        } catch ( Exception ex ) {
            result . setSuccess ( false );
            result . setMessage ( ex . getMessage ());
            sendMns ( akId , akSecret , securityTo ken , result .
toString ());
        }
    }
}

```

```

/**
 * Build a client used for Redis API requests .
 *
 * @ param akId
 * @ param akSecret
 * @ param securityTo ken
 * @ return
 */
private IAcsClient buildClient ( String akId , String
akSecret , String securityTo ken ) {
    IClientProfile clientProfile = DefaultProfile .
getProfile ( REGION_ID , akId , akSecret ,
securityTo ken );
    return new DefaultAcsClient ( clientProfile );
}

/**
 * Send the execution result to Message Service as
a message .
 *
 * @ param ak
 * @ param aks
 * @ param securityTo ken
 * @ param msg
 */
private void sendMns ( String ak , String aks , String
securityTo ken , String msg ) {
    MNSClient client = null ;
    try {
        CloudAccount account = new CloudAccount ( ak ,
aks ,
String . format ( MNS_END_POINT , USER_ID ,
REGION_ID ), securityTo ken );
        client = account . getMNSClient ();
        CloudQueue queue = client . getQueueRe f (
QUEUE_NAME );
        Message message = new Message ();
        message . setMessageBody ( msg );
        queue . putMessage ( message );
    } finally {
        if ( client != null ) {
            client . close ();
        }
    }
}

/**
 * Filter out the private IP addresses that need
to be added to the whitelist of the Redis instance
.
 *
 * @ Param privateIps Filter out the private IP
addresses of previous ECS instances .
 * @ param client
 * @ return
 * @ throws ClientException
 */
private List < String > filterPrivateIpsForAppend ( List <
String > privateIps , IAcsClient client )
throws ClientException {
    List < String > needAppendIps = new ArrayList <>();
    if ( CollectionUtils . isEmpty ( privateIps ) ) {
        return needAppendIps ;
    }
}

```

```

        DescribeSecurityIpsRequest request = new
DescribeSecurityIpsRequest();
        request.setInstanceId ( REDIS_ID );
        DescribeSecurityIpsResponse response = client .
getAcsResponse ( request );
        List<DescribeSecurityIpsResponse.SecurityIpGroup>
securityIpGroups = response
            .getSecurityIpGroups();
        if ( CollectionUtils.isEmpty ( securityIpGroups )) {
            return privateIps ;
        }
        for ( DescribeSecurityIpsResponse.SecurityIpGroup
securityIpGroup : securityIpGroups ) {
            if (! securityIpGroup.getSecurityIpGroupName
()).equals ( DEFAULT_SECURITY_GROUP_NAME )) {
                continue ;
            }
            String securityIps = securityIpGroup .
getSecurityIpList();
            if ( securityIps == null ) {
                continue ;
            }
            String[] securityIpList = securityIps . split
(",");
            List<String> existIps = Arrays.asList (
securityIpList );
            if ( CollectionUtils.isEmpty ( existIps )) {
                continue ;
            }
            for ( String ip : privateIps ) {
                if (! existIps.contains ( ip )) {
                    needAppendIps.add ( ip );
                }
            }
        }
        return privateIps ;
    }

    /**
     * Modify the private IP addresses in the default
group of the Redis instance whitelist .
     *
     * @param client
     * @param needAppendIps
     * @throws ClientException
     */
    private void modifySecurityIps ( IAcscClient client ,
List<String> needAppendIps )
        throws ClientException {
        if ( CollectionUtils.isEmpty ( needAppendIps )) {
            return ;
        }
        ModifySecurityIpsRequest request = new ModifySecu
rityIpsRequest();
        request.setInstanceId ( REDIS_ID );
        String ip = StringUtils.join ( needAppendIps .
toArray (), ",");
        request.setSecurityIps ( ip );
        request.setSecurityIpGroupName ( DEFAULT_SE
CURITY_GROUP_NAME );
        request.setModifyMode ( MODIFY_MODE_APPEND );
        client.getAcsResponse ( request );
    }

```

```

/**
 * Obtain the input content and perform Base64
 decoding .
 *
 * @ param   inputStrea m
 * @ return
 * @ throws  IOExceptio n
 */
private String readInput ( InputStrea m   inputStrea m )
throws  IOExceptio n {
    try {
        byte [] bytes = new byte [ MAX_BYTE_L ENGT H ];
        int tmp ;
        int len = 0 ;
        // Repeatedly read all content .
        while (( tmp = inputStrea m . read () ) != - 1 &&
len < MAX_BYTE_L ENGT H ) {
            bytes [ len ] = ( byte ) tmp ;
            len ++;
        }
        inputStrea m . close ();
        byte [] act = new byte [ len ];
        System . arraycopy ( bytes , 0 , act , 0 , len );
        return new String ( Base64 . decodeBase 64 ( act ),
CHAR_SET );
    } finally {
        inputStrea m . close ();
    }
}

/**
 * Obtain the private IP addresses of all listed
 instances and limit the maximum number of instances
 to 100 per request .
 *
 * @ param   instanceId s The list of instances .
 * @ param   client The request client .
 * @ return
 * @ throws  Exception
 */
public List < String > getInstanc esPrivateI ps ( List <
String > instanceId s , IAcsClient client )
throws Exception {
    List < String > privateIps = new ArrayList <>();
    if ( Collection Utils . isEmpty ( instanceId s )) {
        return privateIps ;
    }
    int size = instanceId s . size ();
    int queryNumbe rPerTime = 100 ;
    int batchCount = ( int ) Math . ceil (( float ) size /
( float ) queryNumbe rPerTime );
    // support 100 instance
    for ( int i = 1 ; i <= batchCount ; i ++ ) {
        int fromIndex = queryNumbe rPerTime * ( i - 1 );
        int toIndex = Math . min ( queryNumbe rPerTime * i
, size );
        List < String > subList = instanceId s . subList (
fromIndex , toIndex );
        DescribeIn stancesReq uest request = new
DescribeIn stancesReq uest ();
        request . setInstanc eIds ( JSON . toJSONStri ng (
subList ));
        DescribeIn stancesRes ponse response = client .
getAcsResp onse ( request );
    }
}

```

```

        List<DescribeInstancesResponse.Instance>
instances = response.getInstances();
        if (CollectionUtils.isEmpty(instances)) {
            continue;
        }
        for (DescribeInstancesResponse.Instance
instance : instances) {
            String privateIp = getPrivateIp(instance);
            if (privateIp != null) {
                privateIps.add(privateIp);
            }
        }
        return privateIps;
    }

    /**
     * Obtain the private IP addresses from DescribeInstancesResponse.Instance.
     *
     * @param instance DescribeInstancesResponse.Instance
     */
    private String getPrivateIp(DescribeInstancesResponse.Instance instance) {
        String privateIp = null;
        if (VPC_NETWORK.equalsIgnoreCase(instance.getNetworkType())) {
            DescribeInstancesResponse.Instance.VpcAttributes vpcAttributes = instance.getVpcAttributes();
            if (vpcAttributes != null) {
                List<String> privateIpAddress = vpcAttributes.getPrivateIpAddress();
                if (!CollectionUtils.isEmpty(privateIpAddress)) {
                    privateIp = privateIpAddress.get(0);
                }
            }
        } else {
            List<String> innerIpAddress = instance.getInnerIpAddress();
            if (!CollectionUtils.isEmpty(innerIpAddress)) {
                privateIp = innerIpAddress.get(0);
            }
        }
        return privateIp;
    }
}

```

Example of `FCResult` . java :

```

package model;

import com.alibaba.fastjson.JSON;

public class FCResult {

    private boolean success = true;

    private String lifecycleHookId;

    private String lifecycleActionToken;
}

```

```

    private String message ;

    public boolean isSuccess () {
        return success ;
    }

    public void setSuccess ( boolean success ) {
        this . success = success ;
    }

    public String getLifecycleHookId () {
        return lifecycleHookId ;
    }

    public void setLifecycleHookId ( String lifecycleHookId )
    {
        this . lifecycleHookId = lifecycleHookId ;
    }

    public String getLifecycleActionToken () {
        return lifecycleActionToken ;
    }

    public void setLifecycleActionToken ( String lifecycleActionToken ) {
        this . lifecycleActionToken = lifecycleActionToken ;
    }

    public String getMessage () {
        return message ;
    }

    public void setMessage ( String message ) {
        this . message = message ;
    }

    @Override
    public String toString () {
        return JSON . toJSONString ( this );
    }
}

```

Example of *HookModel . java* :

```

package model ;

import java . util . List ;

public class HookModel {

    private String lifecycleHookId ;

    private String lifecycleActionToken ;

    private String lifecycleHookName ;

    private String scalingGroupId ;

    private String scalingGroupName ;

    private String lifecycleTransition ;
}

```

```

private String defaultResult ;

private String requestId ;

private String scalingActivityId ;

private List < String > instanceIds ;

public String getLifecycleHookId () {
    return lifecycleHookId ;
}

public void setLifecycleHookId ( String lifecycleHookId
) {
    this . lifecycleHookId = lifecycleHookId ;
}

public String getLifecycleActionToken () {
    return lifecycleActionToken ;
}

public void setLifecycleActionToken ( String
lifecycleActionToken ) {
    this . lifecycleActionToken = lifecycleActionToken ;
}

public String getLifecycleHookName () {
    return lifecycleHookName ;
}

public void setLifecycleHookName ( String lifecycleH
ookName ) {
    this . lifecycleHookName = lifecycleHookName ;
}

public String getScalingGroupId () {
    return scalingGroupId ;
}

public void setScalingGroupId ( String scalingGro
upId )
{
    this . scalingGroupId = scalingGroupId ;
}

public String getScalingGroupName () {
    return scalingGroupName ;
}

public void setScalingGroupName ( String scalingGro
upName ) {
    this . scalingGroupName = scalingGroupName ;
}

public String getLifecycleTransition () {
    return lifecycleTransition ;
}

public void setLifecycleTransition ( String lifecycleT
ransition ) {
    this . lifecycleTransition = lifecycleTransition ;
}

public String getDefaultResult () {
    return defaultResult ;
}

```

```

    }

    public void setDefaultResult ( String defaultResult ) {
        this . defaultResult = defaultResult ;
    }

    public String getRequest Id () {
        return requestId ;
    }

    public void setRequest Id ( String requestId ) {
        this . requestId = requestId ;
    }

    public String getScaling ActivityId () {
        return scalingAct ivityId ;
    }

    public void setScaling ActivityId ( String scalingAct
ivityId ) {
        this . scalingAct ivityId = scalingAct ivityId ;
    }

    public List < String > getInstance Ids () {
        return instanceId s ;
    }

    public void setInstance Ids ( List < String > instanceId s
) {
        this . instanceId s = instanceId s ;
    }
}

```

Example of *MnsMessage Model . java* :

```

package model ;

public class MnsMessage Model {

    private String      userId ;
    private String      regionId ;
    private String      resourceAr n ;
    private HookModel   content ;

    public String getUserId () {
        return userId ;
    }

    public void setUserId ( String userId ) {
        this . userId = userId ;
    }

    public String getRegionI d () {
        return regionId ;
    }

    public void setRegionI d ( String regionId ) {
        this . regionId = regionId ;
    }
}

```

```
public String getResourceArn () {
    return resourceArn ;
}

public void setResourceArn ( String resourceArn ) {
    this . resourceArn = resourceArn ;
}

public HookModel getContent () {
    return content ;
}

public void setContent ( HookModel content ) {
    this . content = content ;
}
}
```