

# Alibaba Cloud

## Auto Scaling Best practices









Document Version: 20200825

## Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

# Document conventions

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

# Table of Contents

1. Build a scalable web application .....	05
2. Use Auto Scaling to reduce costs .....	08
3. Deploy a high-availability compute cluster .....	12
4. Best practices for lifecycle hook and OOS template .....	14
5. Automatically add ECS instances to or remove them from th... ..	15
6. Automatically deploy applications on ECS instances created ... ..	17
7. Use user data to automatically configure ECS instances .....	21
8. Set parameters of scaling configurations to customize ECS in... ..	24
9. Reduce the cost by configuring a cost optimization policy .....	30
10. Use the Alibaba Cloud ESS SDK to create a multi-zone scal... ..	33

# 1. Build a scalable web application

This topic describes how to build a scalable web application by using Auto Scaling that can automatically respond to increases and decreases in business activities. This allows you to handle daily business and traffic spikes during major activities.

## Prerequisites

- Before you perform the operations provided in the tutorial, you must have registered an Alibaba Cloud account. To create an Alibaba Cloud account, click [Create a new Alibaba Cloud account](#).
- A custom image is created for an ECS instance. For more information, see [Create a custom image from an instance](#).

## Scenarios

An e-commerce platform launches promotions during holidays, member days, and shopping festivals to attract users. To handle the traffic spikes during promotions, the operations and maintenance (O&M) personnel estimate the compute resources required for new promotional activities based on historical data. If unexpected traffic spikes occur during peak hours, the O&M personnel must manually create ECS instances. This is time-consuming and may affect the availability of your application.

You can adopt the solutions provided in this topic if your application has the following characteristics:

- Deployed in a cluster that has at least one server.
- Has traffic spikes for a short duration. For example, the traffic spikes last no more than nine hours each day, and no more than 20 days each month.

## Solutions

Auto Scaling automatically scales compute resources based on increases and decreases in business activities without the need for prediction and manual intervention. This ensures the availability of your application. Especially during big promotions such as Double 11, Auto Scaling can deliver up to thousands of ECS instances within minutes, and respond to traffic spikes automatically and timely to ensure service availability.

You can adopt the following solutions:

- Purchase subscription ECS instances to meet daily business requirements.
- Use Auto Scaling to monitor load changes and automatically create ECS instances in response to unexpected traffic spikes.

## Benefits

Auto Scaling enables you to respond to traffic spikes and offers the following benefits:

- Zero backup resource cost

Auto Scaling automatically creates and releases ECS instances based on your requirements. You do not need to maintain backup resources. You only need to reserve compute resources for daily business traffic.

- Zero maintenance cost

You can configure the scaling policy in advance. When the load increases, Auto Scaling automatically creates and adds ECS instances to the whitelist of the ApsaraDB for RDS instance and SLB backend server group. When the load decreases, Auto Scaling automatically removes ECS instances from the SLB backend server group and the whitelist of the ApsaraDB for RDS instance, and then releases the instances. The whole process is automatically triggered and completed without the need for manual intervention.

- **Flexibility and intelligence**

Auto Scaling provides a variety of scaling modes. You can select a combination of multiple scaling modes based on business changes to implement the optimal match for your business. For example, if your web application that requires a large and steady volume of traffic experiences a temporary traffic spike, you can use the dynamic mode based on CloudMonitor metrics. This allows you to monitor average CPU utilization and automatically respond to traffic changes in a timely manner.

## Procedure

Evaluate business modules based on your business architecture and perform the following operations to implement automatic scaling for specified business modules:

- [Step 1: Use a custom image to create subscription ECS instances](#)
- [Step 2: Create and enable a scaling group](#)
- [Step 3: Add subscription ECS instances and configure the automatic scaling policy](#)

### Step 1: Use a custom image to create subscription ECS instances

Create and add the specified number of subscription ECS instances to a scaling group in response to daily traffic requirements of business modules. Perform the following operations:

1. Log on to the [ECS console](#).
2. In the left-side navigation pane, choose **Instances & Images > Images**.
3. In the top navigation bar, select a region.
4. Find the custom image of the web application and click **Create Instance** in the **Actions** column.
5. Configure the parameters to create the instance.
  - Set **Billing Method** to **Subscription**.
  - Information in the **Region** and **Image** sections is automatically filled.

Configure other parameters based on your needs. For more information, see [Create an instance by using the provided wizard](#).

### Step 2: Create and enable a scaling group

Create a scaling group for business modules that require elastic scaling. Select a custom image for the scaling configuration to ensure that automatically created ECS instances meet web application requirements. Perform the following operations:

1. Log on to the [Auto Scaling console](#).
2. In the top navigation bar, select a region.
3. Create a scaling group:
  - Set **Source Type** to **Create from Scratch**.

- Set **Minimum Number of Instances** to 0.
- Set **Network Type** to VPC.
- Set **Multi-zone Scaling Policy** to **Balanced Distribution Policy**.
- Set **Instance Reclaim Mode** to **Release Mode**.
- Bind the SLB and ApsaraDB for RDS instances used by your current business modules.

Configure other parameters based on your needs. For more information, see [Create a scaling group](#).

4. Click **View Scaling Group Details**.

5. Go to the **Instance Configuration Source** page to create a scaling configuration. Set **Image** to the custom image of the web application.

Configure other parameters based on your needs. For more information, see [创建伸缩配置](#).

6. Enable the scaling configuration and scaling group.

### Step 3: Add subscription ECS instances and configure the automatic scaling policy

Add subscription ECS instances to a scaling group and create a target tracking rule to implement automatic scaling based on traffic changes in response to traffic spikes. Perform the following operations:

1. Go to the **ECS Instances** page, and add existing subscription ECS instances to the scaling group.
2. Switch the subscription ECS instances to the **Protected** state to ensure service availability during daily business.
3. Go to the **Basic Information** page, and modify the minimum and maximum numbers of instances in the scaling group based on business needs.
4. Go to the **Scaling Rules** page, and create a target tracking rule.
  - Set **Rule Type** to **Target Tracking Scaling Rule**.
  - Set **Metric Name** to **Average CPU Usage**.
  - Set **Target Value** to 50%.

Configure other parameters based on your needs. For more information, see [Create a scaling rule](#).

## Result

The state of subscription ECS instances is switched to **Protected** to ensure service availability during daily business. The ECS instances in the **Protected** state cannot be removed from the scaling group and their weights in SLB are not affected.

The scaling group automatically keeps the average CPU utilization of ECS instances at about 50%. When the average CPU utilization exceeds 50%, Auto Scaling automatically creates ECS instances to balance loads. When the average CPU utilization drops below 50%, Auto Scaling automatically releases ECS instances to reduce costs. The number of ECS instances remains greater than or equal to the specified minimum number of instances, and less than or equal to the maximum number of instances to meet business requirements and keep costs within expectation.

## 2. Use Auto Scaling to reduce costs

This topic describes how to use Auto Scaling to purchase pay-as-you-go and preemptible ECS instances to reduce costs during predictable business peaks.

### Prerequisites

- Before you perform the operations provided in the tutorial, you must have registered an Alibaba Cloud account. To create an Alibaba Cloud account, click [Create a new Alibaba Cloud account](#).
- A custom image is created for an ECS instance. For more information, see [Create a custom image from an instance](#).

### Scenarios

An online education platform experiences traffic peaks from 17:00 to 22:00 every day. However, during other times of the day, the business traffic is significantly lower. To ensure that the platform can deliver reliable services during peak hours, the number of compute resources is scaled based on the peak traffic loads. During off-peak hours, these resources are idle, which results in a large amount of wasted cost. Furthermore, when the platform experiences unexpected traffic spikes, ECS instances must be manually created to ensure service availability.

You can adopt the solutions provided in this topic if your application has the following characteristics:

- Deployed in a cluster that has at least one server.
- Has predictable traffic patterns. For example, traffic peaks occur from 17:00 to 22:00 each day and the compute resources are idle during other times of the day.

### Solutions

Auto Scaling uses a combination of pay-as-you-go and preemptible instances to meet peak traffic requirements at lower costs.

You can adopt the following solutions:

- Purchase subscription ECS instances to maintain a baseline compute capability for off-peak hours.
- Specify multiple instance types and use a combination of pay-as-you-go and preemptible instances to scale compute capabilities for peak hours. Auto Scaling creates ECS instances based on unit prices of vCPUs in ascending order. Instances that use lowest-priced vCPUs are preferentially created.

### Benefits

Auto Scaling enables you to reduce costs and offers the following benefits:

- Zero backup resource cost

Auto Scaling automatically creates and releases ECS instances based on your requirements. You do not need to maintain backup resources. You only need to reserve compute resources for off-peak hours.

- Zero maintenance cost



You can configure the scaling policy in advance. When the load increases, Auto Scaling automatically creates and adds ECS instances to the whitelist of the ApsaraDB for RDS instance and SLB backend server group. The whole process is automatically triggered and completed without the need for manual intervention.

- High cost-effectiveness

Auto Scaling supports the combination of pay-as-you-go and preemptible instances. You can purchase ECS instances at up to 90% discount. If the preemptible instances are insufficient, pay-as-you-go instances are created to ensure service availability. The cost optimization policy also supports supplemental preemptible instances. After this feature is enabled, Auto Scaling automatically creates preemptible instances at lowest price five minutes before existing preemptible instances are released.

## Procedure

Evaluate business modules based on your business architecture and perform the following operations to reduce costs for required business modules:

- **Step 1: Use a custom image to create subscription ECS instances**
- **Step 2: Create and enable a scaling group**
- **Step 3: Add subscription ECS instances and configure the automatic scaling policy**

### Step 1: Use a custom image to create subscription ECS instances

Create and add the specified number of subscription ECS instances to a scaling group in response to off-peak traffic requirements of business modules. Perform the following operations:

1. Log on to the [ECS console](#).
2. In the left-side navigation pane, choose **Instances & Images > Images**.
3. In the top navigation bar, select a region.
4. Find the custom image of the application and click **Create Instance** in the **Actions** column.
5. Configure the parameters to create an instance.
  - Set **Billing Method** to **Subscription**.
  - Information in the **Region** and **Image** sections is automatically filled.

Configure other parameters based on your needs. For more information, see [Create an instance by using the provided wizard](#).

### Step 2: Create and enable a scaling group

Create a scaling group for business modules that require lower costs. Select a custom image for the scaling configuration to ensure that automatically created ECS instances meet application requirements. Perform the following operations:

1. Log on to the [Auto Scaling console](#).
2. In the top navigation bar, select a region.
3. Create a scaling group.
  - Set **Source Type** to **Create from Scratch**.
  - Set **Minimum Number of Instances** to **0**.
  - Set **Network Type** to **VPC**.

- Set **Multi-zone Scaling Policy** to **Cost Optimization Policy**.
  - Set **Minimum Pay-as-you-go Instances** to **0**.
  - Set **Percentage of Pay-as-you-go Instances** to **30%**.
  - Set **Lowest Cost Instance Types** to **3**.
  - Enable the supplemental preemptible instances mode.
- Set **Reclaim Mode** to **Release Mode**.
- Bind the SLB and ApsaraDB for RDS instances used by your current business modules.

Configure other parameters based on your needs. For more information, see [Create a scaling group](#).

4. Click **View Scaling Group Details**.

5. Go to the **Instance Configuration Source** page to create a scaling configuration.

- Set **Billing Method** to **Preemptible Instance**.
- Select at least three instance types.
- Set **Image** to your custom image.

Configure other parameters based on your needs. For more information, see [创建伸缩配置](#).

6. Enable the scaling configuration and scaling group.

### Step 3: Add subscription ECS instances and configure the automatic scaling policy

Add subscription ECS instances to a scaling group and create a step scaling rule to implement automatic and smooth scaling based on business changes. You can significantly reduce costs by using a combination of subscription and preemptible instances. Perform the following operations:

1. Go to the **ECS Instances** page, and add existing subscription ECS instances to the scaling group.
2. Switch the subscription ECS instances to the **Protected** state to ensure service availability during off-peak hours.
3. Go to the **Basic Information** page, and modify the minimum and maximum numbers of instances in the scaling group based on business needs.
4. Go to the **Scaling Rules** page, and create a step scaling rule.
  - Set **Rule Type** to **Step Scaling Rule**.
  - Set **Monitoring Type** to **System Monitoring**.
  - Set **Run At** to the time when the average CPU utilization is greater than 50% for three consecutive times.
  - Set **Operation** based on the following rules:
    - Add five instances when the average CPU utilization is greater than or equal to 60% and less than 70%.
    - Add 10 instances when the average CPU utilization is greater than or equal to 70%.

Configure other parameters based on your needs. For more information, see [Create a scaling rule](#).

## Result

The state of subscription ECS instances is switched to Protected to ensure service availability during off-peak hours. The ECS instances in the Protected state cannot be removed from the scaling group, and their weights in SLB are not affected.

During peak hours, Auto Scaling automatically creates a specific number of ECS instances based on the average CPU utilization to implement smooth scaling. Due to the cost optimization policy and supplemental preemptible instances mode, you can purchase ECS instances at lower costs.

# 3. Deploy a high-availability compute cluster

This topic describes how to use Auto Scaling to evenly distribute ECS instances across zones and deploy a high-availability compute cluster at lower costs by using preemptible ECS instances.

## Prerequisites

- Before you perform the operations provided in the tutorial, you must have registered an Alibaba Cloud account. To create an Alibaba Cloud account, click [Create a new Alibaba Cloud account](#).
- A custom image is created for an ECS instance. For more information, see [Create a custom image from an instance](#).

## Scenarios

An online advertising provider uses machine learning to implement targeted advertising. During peak hours, the provider requires a large number of compute resources. This results in higher costs and may face scalability problems, such as insufficient resources, insufficient time to manually create ECS instances, and service disruption. All these problems pose risks to the business.

You can adopt the solutions provided in this topic if your application is applicable to the following scenarios:

- Distributed big data computing
- Artificial intelligence training

## Solutions

Auto Scaling can provision a compute cluster in a short amount of time. The balanced distribution policy allows you to automatically distribute compute nodes across multiple zones. Auto Scaling also performs health checks on ECS instances to ensure the high availability of the compute cluster.

You can adopt the following solutions:

- Use Auto Scaling to distribute compute nodes across multiple zones and specify multiple instance types.
- Purchase preemptible ECS instances to reduce costs.

## Benefits

Auto Scaling enables you to deploy a high-availability compute cluster and offers the following benefits:

- Zero maintenance cost

You can configure the scaling policy in advance. When the load increases, the scaling group automatically creates and adds ECS instances to the whitelist of the ApsaraDB for RDS instance. When the load decreases, the scaling group automatically removes ECS instances from the whitelist of the ApsaraDB for RDS instance, and then releases the instances. The whole process is automatically triggered and completed without the need for manual intervention.

- **High cost-effectiveness**

Auto Scaling supports preemptible ECS instances. You can purchase preemptible instances at up to 90% discount.

- **High availability**

Auto Scaling uses the balanced distribution policy to automatically distribute and deploy compute nodes across zones. This ensures service availability and reduces the risk that resources in a zone may be insufficient. Auto Scaling automatically performs health checks to ensure the availability of ECS instances in a scaling group.

## Procedure

Evaluate business modules based on your business architecture and create scaling groups for the business modules that require high-availability clusters. Select a custom image for the scaling configuration to ensure that the automatically created ECS instances meet application requirements.

1. Log on to the [Auto Scaling console](#).
2. In the top navigation bar, select a region.
3. Create a scaling group.
  - Set **Source Type** to **Create from Scratch**.
  - Set **Minimum Number of Instances** to **100**.
  - Set **Network Type** to **VPC**.
  - Select VSwitches across multiple zones.
  - Set **Multi-zone Scaling Policy** to **Balanced Distribution Policy**.
  - Bind the ApsaraDB for RDS instances used by your current business modules.

Configure other parameters based on your needs. For more information, see [Create a scaling group](#).

4. Click **View Scaling Group Details**.
5. Go to the **Instance Configuration Source** page to create a scaling configuration.
  - Set **Billing Method** to **Preemptible Instance**.
  - Set **Image** to your custom image.

Configure other parameters based on your needs. For more information, see [创建伸缩配置](#).

6. Enable the scaling configuration and scaling group.

## Result

After the scaling group is enabled, the scaling group automatically distributes 100 ECS instances evenly across the selected zones. This can reduce impacts on the application when a zone has insufficient resources. The scaling group automatically creates new preemptible instances after the previous preemptible instances are reclaimed. Additionally, the scaling group automatically removes unhealthy ECS instances and creates new ECS instances. This ensures the high availability of clusters and also reduces costs.

## 4. Best practices for lifecycle hook and OOS template

# 5. Automatically add ECS instances to or remove them from the whitelist of a Redis instance

This topic describes how to use a lifecycle hook to add ECS instances to or remove them from the whitelist of an ApsaraDB for Redis (Redis) instance.

Auto Scaling lifecycle hook whitelist of a Redis instance OOS

## Context

A scaling group can be associated with Server Load Balancer (SLB) or ApsaraDB for RDS (RDS) instances, but cannot be associated with Redis instances. If your business data is stored in a Redis instance, you must manually add your ECS instances to or remove them from the whitelist of the Redis instance. This is time-consuming and inefficient. Operation Orchestration Service (OOS) is an automated operations service provided by Alibaba Cloud. You can use lifecycle hooks of Auto Scaling and OOS templates to automatically add ECS instances to or remove them from the whitelist of a Redis instance.

## Procedure

1. Log on to the [Auto Scaling console](#).
2. In the top navigation bar, select a region.
3. Find the target scaling group and use one of the following methods to open the details page of the scaling group:
  - Click the ID of the scaling group in the **Scaling Group Name/ID** column.
  - Click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **Lifecycle Hooks**.
5. Click **Create Lifecycle Hook**.
6. Configure parameters that are related to the whitelist of a Redis instance.
  - i. In the **Notification Method** section, select **OOS Template** and then select **Public Templates**.
  - ii. Select **ACS-ESS-LifeCycleModifyRedisIPWhitelist** from the public template list. You can enter the keyword **redis** to search for the template.
  - iii. Click **Configure Parameters**.

## iv. Configure parameters for the OOS template.

Parameter	Description
<code>dbInstanceId</code>	The ID of the Redis instance.
<code>modifyMode</code>	The specified operation type. It must be the same as that of the scaling activity. <ul style="list-style-type: none"><li>▪ <b>Append:</b> corresponds to the scale-out event, and specifies that the internal IP addresses of ECS instances are automatically added to the whitelist of the Redis instance.</li><li>▪ <b>Delete:</b> corresponds to the scale-in event, and specifies that the internal IP addresses of ECS instances are automatically removed from the whitelist of the Redis instance.</li></ul>
<code>Permissions</code>	The permission to automatically add ECS instances to or remove them from the whitelist of the Redis instance. Default value: <code>OOSServiceRole</code> . You can create other RAM roles. For more information, see <a href="#">Grant RAM permission for OOS</a> .

7. Click **OK**.

8. Configure other required parameters for the lifecycle hook and click **Create**. For more information about other required parameters for a lifecycle hook, see [Create a lifecycle hook](#).



# 6. Automatically deploy applications on ECS instances created by Auto Scaling

This topic uses CentOS as an example to describe how to use a Shell script to automatically deploy applications on ECS instances created by Auto Scaling.

## Context

CentOS can be booted into the following runlevels:

- Runlevel 0: the halt runlevel.
- Runlevel 1: causes the system to start up in a single user mode under which only the root user can log on.
- Runlevel 2: boots the system into the multi-user mode with text-based console logon capability. This runlevel does not start the network.
- Runlevel 3: boots the system into the multi-user mode with text-based console logon capability. This runlevel starts the network.
- Runlevel 4: undefined runlevel. This runlevel can be configured to provide a custom boot state.
- Runlevel 5: boots the system into the multi-user mode. This runlevel starts the graphical desktop environment at the end of the boot process.
- Runlevel 6: reboots the system.

You can use a script to automatically install or update applications or run specific code on ECS instances created by Auto Scaling. To do so, add the script to a custom image and configure a command to run the script when the operating system boots. Then, select the custom image in a scaling configuration. After an ECS instance is created based on the scaling configuration, the script is automatically run on the ECS instance.

CentOS 6 and earlier versions use System V init as the initialization system, whereas CentOS 7 uses Systemd as the initialization system. System V init and Systemd are quite different in the ways that they operate. This topic describes how to configure a script in CentOS 6 and earlier versions and in CentOS 7 respectively.

## CentOS 6 and earlier versions

This section describes how to configure a script in CentOS 6 and earlier versions.


1. Create a Shell script for testing.

```
#!/bin/sh
# chkconfig: 6 10 90
# description: Test Service
echo "hello world!"
```

The `CHKCONFIG` command in the preceding script sets the runlevel and priorities for running the script when the operating system boots and shuts down. The value 6 indicates runlevel 6, which means that the script is run when the operating system reboots. For more information about runlevels, see the background information in this topic. The value 10 indicates the priority for running the script when the operating system boots. The value 90 indicates the priority for running the script when the operating system shuts down. A priority ranges from 0 to 100, where a higher value indicates a lower priority.

 **Note** To make sure that the ECS instance is released only after the script is run on the ECS instance, change runlevel 6 to runlevel 0 in the preceding script.

2. Place the test script in the `/etc/rc.d/init.d/` directory and run the `chkconfig --level 6 test on` command. Then, the script is run each time the operating system reboots.

 **Note** To make sure that the ECS instance is released only after the script is run on the ECS instance, change runlevel 6 to runlevel 0 in the preceding script. Then, the script is run each time the operating system shuts down.

For example, you can use the following sample script to automatically install PHPWind. You still need to enter the password for logging on to the database. Modify the script as required in actual use.

```
cd /tmp
echo "phpwind"
yum install -y \
unzip \
wget \
httpd \
php \
php-fpm \
php-mysql \
php-mbstring \
php-xml \
php-gd \
php-pear \
php-devel
chkconfig php-fpm on \
&& chkconfig httpd on
wget http://pwfiles.oss-cn-hangzhou.aliyuncs.com/com/soft/phpwind_v9.0_utf8.zip \
&& unzip -d pw phpwind_v9.0_utf8.zip \
&& mv pw/phpwind_v9.0_utf8/upload/* /var/www/html \
&& wget http://ess.oss-cn-hangzhou.aliyuncs.com/ossupload_utf8.zip -O ossupload_utf8.zip \
&& unzip -d ossupload ossupload_utf8.zip \
&& /bin/cp -rf ossupload/ossupload_utf8/* /var/www/html/src/extensions/ \
&& chown -R apache:apache /var/www/html
service httpd start && service php-fpm start
echo "Install CloudMonitor"
wget http://update2.aegis.aliyun.com/download/quartz_install.sh
chmod +x quartz_install.sh
bash quartz_install.sh
echo "CloudMonitor installed"
```

## CentOS 7

This section describes how to configure a script in CentOS 7. CentOS 7 uses Systemd as the initialization system. After you configure a script by following the steps in this section, the script can be run when the system is shut down.

1. Create the script to run.
2. Create the *run-script-when-shutdown.service* file in the */etc/systemd/system* directory. Add the following content to the file. Change the value of the *ExecStop* variable to the absolute path of the script to run.

```
[Unit]
Description=service to run script when shutdown
After=syslog.target network.target

[Service]
Type=simple
ExecStart=/bin/true
ExecStop=/path/to/script/to/run
RemainAfterExit=yes

[Install]
WantedBy=default.target
```

3. Run the following commands to start the `run-script-when-shutdown` service:

```
systemctl enable run-script-when-shutdown
systemctl start run-script-when-shutdown
```

 Note

- You can configure the `run-script-when-shutdown` service to specify the script to run. This allows you to flexibly change the script to run.
- If the `run-script-when-shutdown` service is no longer needed, run the `systemctl disable run-script-when-shutdown` command to disable the service.

# 7. Use user data to automatically configure ECS instances

To provide scaling services that are more efficient and flexible, Auto Scaling allows you to configure user data in scaling configurations to customize ECS instances. You can use this feature to automatically configure ECS instances, for example, to automatically install applications on ECS instances. This allows you to scale applications in a more secure and quick manner.

## Prerequisites

Before you perform the operations provided in the tutorial, you must have registered an Alibaba Cloud account. To create an Alibaba Cloud account, click [Create a new Alibaba Cloud account](#).

To verify the effect of user data, you must log on to ECS instances. We recommend that you use Secure Shell (SSH) key pairs to log on to Linux instances. For more information, see [Create an SSH key pair](#) and [Connect to a Linux instance by using an SSH key pair](#).

## Context

This topic uses an example to describe how to use user data in Auto Scaling. You can customize user data based on your business requirements.

For more information about user data, see [Prepare user data](#). Both Windows and Linux instances support user data. You can use user data in the following ways:

- Configure a script that is run when an ECS instance starts. In this way, you can customize the startup behavior of the ECS instance.
- Pass data to an ECS instance. You can reference the data on the ECS instance.

Compared with using open-source IT infrastructure management tools such as Terraform, using user data that is supported natively by Auto Scaling to manage the infrastructure is more efficient and secure. You only need to configure a Base64-encoded custom script and pass the script to a scaling configuration as user data. ECS instances created based on the scaling configuration can run the script during startup to automatically deploy applications. In this way, you can scale applications. When you use user data, note that:

- The network type of the scaling group must be Virtual Private Cloud (VPC).
- The user data must be Base64-encoded.
- We recommend that you do not configure confidential information, such as passwords and keys, in user data because user data is passed to instances in plaintext. If you must pass confidential information, we recommend that you encrypt the confidential information before encoding it in Base64 and decrypt it on the instance.

If you call an API operation to create a scaling configuration, you can configure user data in the `UserData` parameter. For more information, see [CreateScalingConfiguration](#).

In addition to user data, you can use SSH key pairs, Resource Access Management (RAM) roles, and tags to customize ECS instances efficiently and flexibly. For more information, see [Set parameters of scaling configurations to customize ECS instances](#).

## Procedure

Follow these steps to configure user data in a scaling configuration:

1. **Step 1: Prepare user data**
2. **Step 2: Create and enable a scaling group**
3. **Step 3: Verify the user data**

## Step 1: Prepare user data

You can configure a custom Shell script in user data and enable the script to run when ECS instances start. When you customize a Shell script, note that:

- **Format:** The first line must start with `#!`, such as `#!/bin/sh`.
  - **Limit:** The script size cannot exceed 16 KB before the script is Base64 encoded.
  - **Frequency:** The script is run only when instances are started for the first time.
1. Customize a Shell script to configure the Domain Name System (DNS), yum, and Network Time Protocol (NTP) services when an ECS instance starts. The Shell script is as follows:

```
#!/bin/sh
# Modify DNS
echo "nameserver 8.8.8.8" | tee /etc/resolv.conf
# Modify yum repo and update
rm -rf /etc/yum.repos.d/*
touch myrepo.repo
echo "[base]" | tee /etc/yum.repos.d/myrepo.repo
echo "name=myrepo" | tee -a /etc/yum.repos.d/myrepo.repo
echo "baseurl=http://mirror.centos.org/centos" | tee -a /etc/yum.repos.d/myrepo.repo
echo "gpgcheck=0" | tee -a /etc/yum.repos.d/myrepo.repo
echo "enabled=1" | tee -a /etc/yum.repos.d/myrepo.repo
yum update -y
# Modify NTP Server
echo "server ntp1.aliyun.com" | tee /etc/ntp.conf
systemctl restart ntpd.service
```

2. Encode the Shell script in Base64. The Base64-encoded Shell script is as follows:

```
lyEvYmluL3NoCiMgTW9kaWZ5IEROUwplY2hviCJuYW1lc2VydMvYIDguOC44LjgiIHwgdGVlIC9ldGMvcMv
zb2x2LmNvbMvYKIyBNb2Rpb2NkgeXVtIHJlcG8gYW5kIHVwZGF0ZQpybSAtcmYgL2V0Yy95dW0ucMvVwb3
MuZC8qCnRvdWNoIG15cmVwby5yZXBvcmVjaG8giltiYXNlXS IgfCB0ZWUgL2V0Yy95dW0ucMvVwb3MuZC
9teXJlcG8ucMvVwbwplY2hviCJuYW1lPW15cmVwbylGfCB0ZWUgLWEgL2V0Yy95dW0ucMvVwb3MuZC9teX
JlcG8ucMvVwbwplY2hviCJiYXNldXJsPW h0dHA6Ly9taXJyb3luY2VudG9zLm9yZy9jZW50b3MiiHwgdGVlIC1
hIC9ldGMveXVtLnJlcG9zLmQvbXlyZXBvLnJlcG8KZWNoYAiZ3BnY2hlY2s9MCIgfCB0ZWUgLWEgL2V0Yy
95dW0ucMvVwb3MuZC9teXJlcG8ucMvVwbwplY2hviCJlbnFibGVkPTEiIHwgdGVlIC1hIC9ldGMveXVtLnJlcG9
zLmQvbXlyZXBvLnJlcG8KeXVtIHVwZGF0ZSAteQojIE1vZGlmeSBOVFAGU2VydMvYcmVjaG8gInNlcnZlci
BudHAXLmFs aXl1bi5jb20iiHwgdGVlIC9ldGMvbnRwLmNvbMvYKc3lzdGVtY3RsIHJlc3RhcncQgbnRwZC5zZ
XJ2aWNl
```

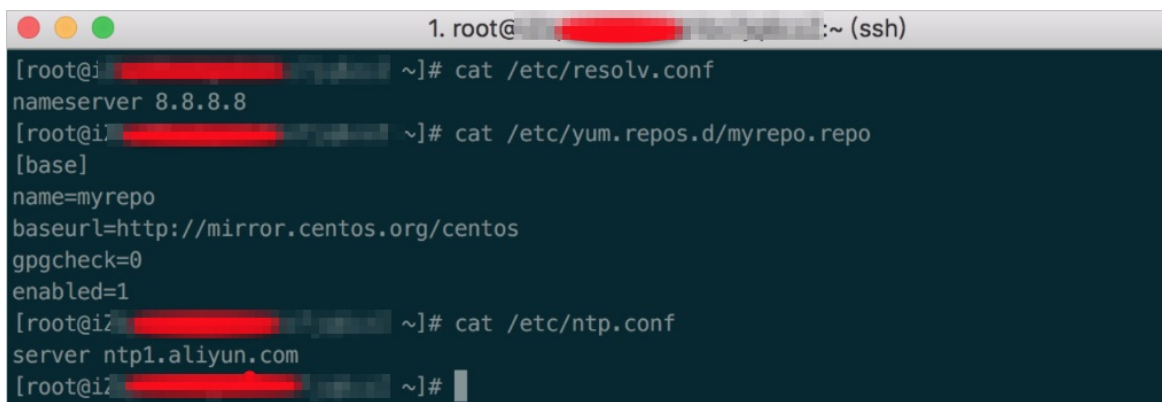
## Step 2: Create and enable a scaling group

1. Create a scaling group. For more information, see [Create a scaling group](#). Note that:
  - **Minimum Instances:** Set this parameter to 1. An ECS instance is created after the scaling group is enabled.
  - **Instance Configuration Source:** Select **Custom Scaling Configuration** for this parameter.
  - **Network Type:** Select **VPC** for this parameter and set the **VPC ID** and **VSwitch** parameters.
2. In the **The scaling group has been created** dialog box that appears, click **Create Now** to create a scaling configuration.
3. Create a scaling configuration. For more information, see [创建伸缩配置](#). Note that:
  - On the **Basic Configurations** page, select **Ubuntu 16.04 64-bit** as the image.
  - On the **System Configurations** page, select the user data prepared in Step 1 and select an existing SSH key pair for logging on to ECS instances created based on this scaling configuration.
4. In the **Created** dialog box, click **Enable Configuration**.
5. In the **Apply Scaling Configuration** dialog box, click **OK**.
6. In the **Enable Scaling Group** dialog box, click **OK**.

## Step 3: Verify the user data

The minimum number of instances in the scaling group is set to 1. Therefore, the scaling group automatically creates an ECS instance after it is enabled.

1. View the scaling activity. For more information, see [View the details of a scaling activity](#).
2. Log on to the instance. For more information, see [Connect to a Linux instance by using an SSH key pair](#).
3. Check the status of services on the instance. The following figure shows that the DNS, yum, and NTP services are enabled on the ECS instance. This indicates that the user data configured in the scaling configuration takes effect.



```
1. root@ [REDACTED]:~ (ssh)
[root@i[REDACTED] ~]# cat /etc/resolv.conf
nameserver 8.8.8.8
[root@i[REDACTED] ~]# cat /etc/yum.repos.d/myrepo.repo
[base]
name=myrepo
baseurl=http://mirror.centos.org/centos
gpgcheck=0
enabled=1
[root@i[REDACTED] ~]# cat /etc/ntp.conf
server ntp1.aliyun.com
[root@i[REDACTED] ~]#
```

# 8. Set parameters of scaling configurations to customize ECS instances

Auto Scaling automatically adds and removes Elastic Compute Service (ECS) instances based on your business requirements. To provide scaling services that are more flexible, Auto Scaling allows you to configure the following settings in scaling configurations to customize ECS instances: tags, Secure Shell (SSH) key pairs, Resource Access Management (RAM) roles, and user data. This topic describes tags, SSH key pairs, RAM roles, and user data, and how to configure them in scaling configurations.

## Prerequisites

Before you perform the operations provided in the tutorial, you must have registered an Alibaba Cloud account. To create an Alibaba Cloud account, click [Create a new Alibaba Cloud account](#).

## Context

Auto Scaling not only can scale ECS instances during peak or off-peak traffic hours, but also can customize the ECS instances that are created. Auto Scaling allows you to set various parameters of scaling configurations to customize ECS instances efficiently and flexibly based on your business requirements.

- Tag

For more information about tags, see [Overview](#). Tags can be used to identify resources and user groups. With tags, enterprises and individuals can categorize their ECS resources to make it easier to search for and aggregate resources. When you create a scaling configuration, you can select the tags to be attached to the ECS instances created based on the scaling configuration.

If you call an API operation to create a scaling configuration, you can specify the tags in the Tags parameter. For more information, see [CreateScalingConfiguration](#).

- SSH key pair

For more information about SSH key pairs, see [SSH key pair overview](#). Alibaba Cloud only supports 2048-bit RSA key pairs. SSH key pairs only apply to Linux instances. After an SSH key pair is created, Alibaba Cloud holds the public key and offers you the private key.

Compared with using a password, using an SSH key pair to log on to a Linux instance is more efficient and secure. You can specify an SSH key pair when creating a scaling configuration. After Auto Scaling creates an ECS instance by using the scaling configuration, the instance stores the public key of the specified SSH key pair. You can use the private key to log on to the ECS instance from your local device.

If you call an API operation to create a scaling configuration, you can specify an SSH key pair in the KeyPairName parameter. For more information, see [CreateScalingConfiguration](#).

- RAM role

Resource Access Management (RAM) is a service provided by Alibaba Cloud for managing user identities and resource access permissions. RAM allows you to create different roles and grant different permissions on Alibaba Cloud services to each role.



For more information about RAM roles, see [Overview](#). ECS instances can assume a RAM role to obtain the permissions granted to the RAM role. When you specify a RAM role in a scaling configuration, make sure that ECS has been set to the trusted entity of the RAM role. Otherwise, Auto Scaling cannot create ECS instances based on the scaling configuration.

If you call an API operation to create a scaling configuration, you can specify a RAM role in the `RamRoleName` parameter. For more information, see [CreateScalingConfiguration](#).

- **User data**

For more information about user data of ECS instances, see [Prepare user data](#). Both Windows and Linux instances support user data. You can use user data in the following ways:

- Configure a script that is run when an ECS instance starts. In this way, you can customize the startup behavior of the ECS instance.
- Pass data to an ECS instance. You can reference the data on the ECS instance.

Compared with using open-source IT infrastructure management tools such as Terraform, using user data that is supported natively by Auto Scaling to manage the infrastructure is more efficient and secure. You only need to configure a Base64-encoded custom script and pass the script to a scaling configuration as user data. ECS instances created based on the scaling configuration can run the script during startup to automatically deploy applications. In this way, you can scale applications. When you use user data, note that:

- The network type of the scaling group must be Virtual Private Cloud (VPC).
- The user data must be Base64-encoded.
- We recommend that you do not configure confidential information, such as passwords and keys, in user data because user data is passed to instances in plaintext. If you must pass confidential information, we recommend that you encrypt the confidential information before encoding it in Base64 and decrypt it on the instance.

If you call an API operation to create a scaling configuration, you can configure user data in the `UserData` parameter. For more information, see [CreateScalingConfiguration](#).

Proper use of Auto Scaling can reduce your server cost, service management cost, and O&M cost. To help you understand and properly use Auto Scaling, this topic demonstrates how to configure the preceding settings in a scaling configuration for Auto Scaling to automatically scale and customize ECS instances. Specifically, this topic demonstrates how to configure tags, an SSH key pair, a RAM role, and user data containing a custom script in a scaling configuration. When an ECS instance is created based on the scaling configuration, the tags are attached to the ECS instance, and the ECS instance assumes the RAM role. You can use the SSH key pair to log on the ECS instance. The custom script is automatically run when the ECS instance starts.

## Procedure

Follow these steps to configure custom settings, including tags, an SSH key pair, a RAM role, and user data, in a scaling configuration:

1. [Step 1: Prepare custom settings](#)
2. [Step 2: Apply the preceding settings](#)
3. [Step 3: Verify the preceding settings](#)

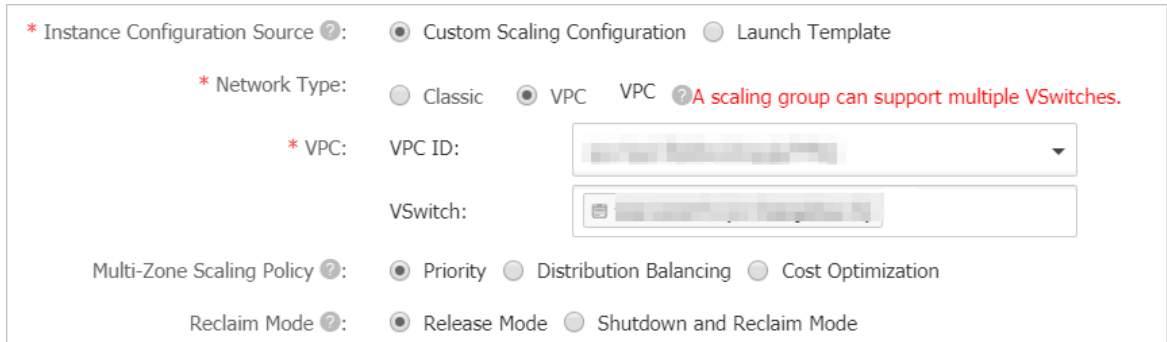
### Step 1: Prepare custom settings

Follow these steps to create tags, an SSH key pair, a RAM role, and user data:



Follow these steps to create a scaling group and a scaling configuration and apply the preceding settings in the scaling configuration:

1. Create a scaling group. For more information, see [Create a scaling group](#). Note that:
  - o **Minimum Instances:** Set this parameter to 1. An ECS instance is created after the scaling group is enabled.
  - o **Instance Configuration Source:** Select **Custom Scaling Configuration** for this parameter.
  - o **Network Type:** Select **VPC** for this parameter and set the **VPC ID** and **VSwitch** parameters.

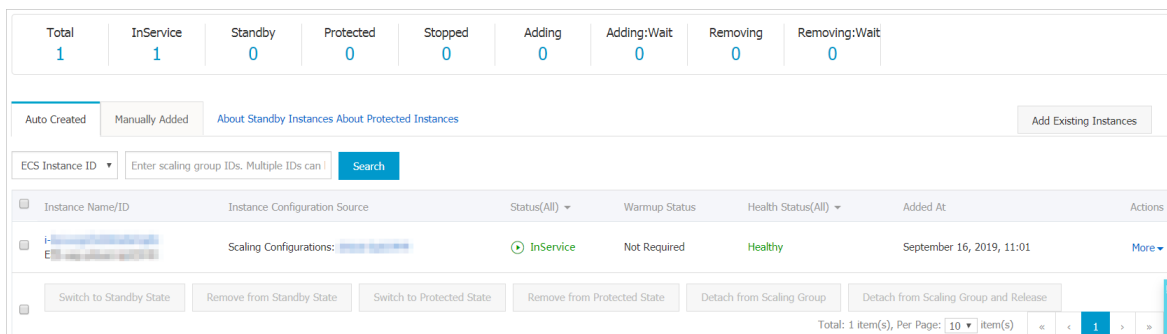


2. In the **The scaling group has been created** dialog box that appears, click **Create Now** to create a scaling configuration.
3. Create a scaling configuration. For more information, see [创建伸缩配置](#). Note that:
  - o On the **Basic Configurations** page, select **Ubuntu 16.04 64-bit** as the image.
  - o On the **System Configurations** page, select the tags, SSH key pair, RAM role, and user data that are created in Step 1.
4. In the **Created** dialog box, click **Enable Configuration**.
5. In the **Apply Scaling Configuration** dialog box, click **OK**.
6. In the **Enable Scaling Group** dialog box, click **OK**.

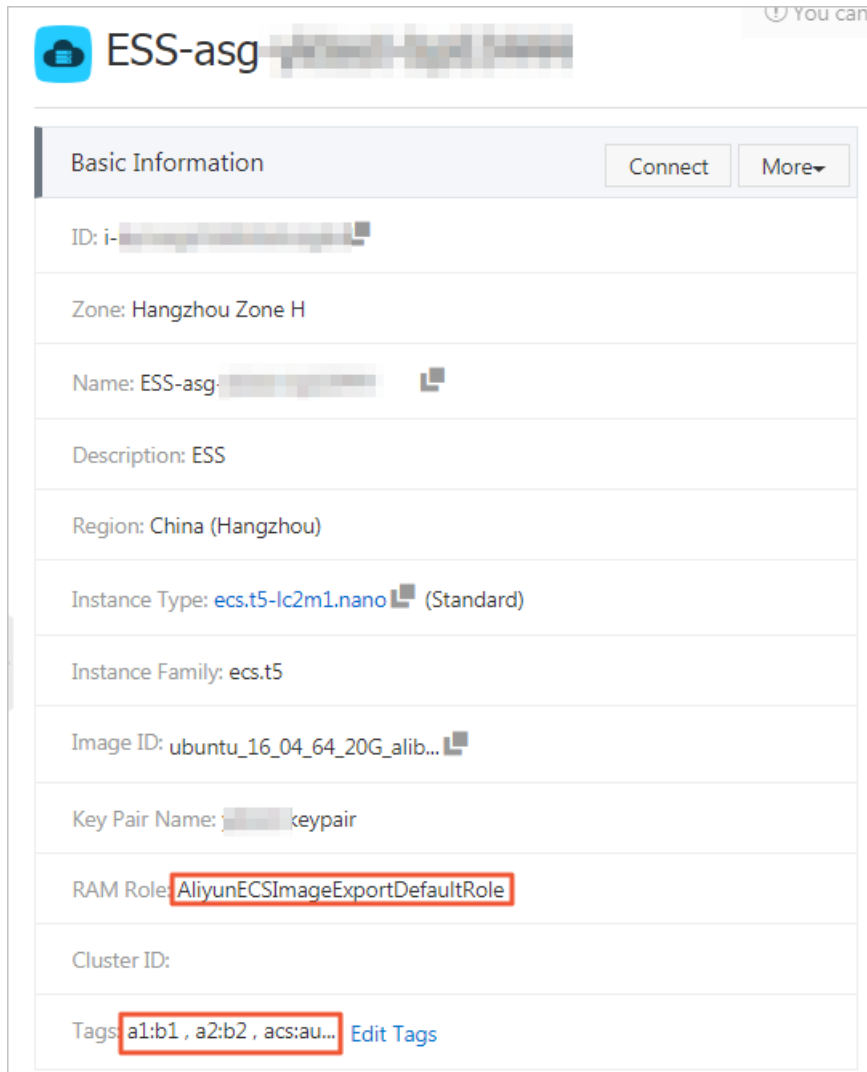
### Step 3: Verify the preceding settings

In Step 2, the minimum number of instances in the scaling group is set to 1. Therefore, the scaling group automatically creates an ECS instance after it is enabled.

1. View the automatically created ECS instance. For more information, see [Query ECS instances](#).



2. In the **Instance Name/ID** column, click the target instance ID to view details of the instance. The following figure shows details of the instance. You can find that the instance has assumed the RAM role and the tags have been attached to the instance.



3. Use the SSH key pair to log on to the instance. For more information, see [Connect to a Linux instance by using an SSH key pair](#). The following figure shows a successful logon. This indicates that the SSH key pair takes effect.

```
Using username "root".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-151-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Welcome to Alibaba Cloud Elastic Compute Service !


root@i-...:~#
```

4. Run the following command to view the content of the `/root/output10.txt` file:

```
cat /root/output10.txt
```

The following figure shows that the file contains the specified string. This indicates that the user data configured in the scaling configuration takes effect.

```
root@i-1234567890:~# cat /root/output10.txt
Hello World. The time is now Mon, 16 Sep 2019 11:01:26 +0800!
root@i-1234567890:~# █
```

 **Note** This topic uses a simple Shell script as an example. You can create a script based on your requirements to customize more startup behaviors.

# 9.Reduce the cost by configuring a cost optimization policy

This topic describes how to configure a cost optimization policy for a scaling group. A cost optimization policy can be used to create multiple types of Elastic Compute Service (ECS) instances in different zones. This increases the chance of successful scaling and reduces the cost.

## Prerequisites

- Before you perform the operations provided in the tutorial, you must have registered an Alibaba Cloud account. To create an Alibaba Cloud account, click [Create a new Alibaba Cloud account](#).
- A Virtual Private Cloud (VPC) is created. For more information, see [Create a VPC](#).
- Multiple VSwitches are created in different zones of the VPC. For more information, see [Create a VSwitch](#).

## Context

Auto Scaling supports creating ECS instances of multiple types in a scaling group. You can specify multiple instance types in a scaling configuration. Auto Scaling creates instances based on the priorities of the instance types. If an instance of the type with the highest priority cannot be created, Auto Scaling creates an instance of the type with the next highest priority. The same rule applies to the rest instance types until an instance is created. This increases the chance of successful scaling when instances of a specified type are insufficient in the inventory. During the business peak periods, ECS instances must be created in time to bear traffic. In this case, the system must focus on creating ECS instances with sufficient performance to meet business requirements, instead of creating ECS instances of a specified type. We recommend that you specify multiple instance types to guarantee that instances can be created.

Auto Scaling supports creating ECS instances in different zones. You can select VSwitches in multiple zones when you create a scaling group. If the zone where a VSwitch resides does not have sufficient ECS instances in stock, Auto Scaling attempts to create instances in other zones. This increases the chance of successful scaling. After configuring multiple zones, you can also configure a multi-zone scaling policy based on your actual business needs. Multi-zone scaling policies include priority policies, distribution balancing policies, and cost optimization policies.

### Note

- Multi-zone scaling policies are available only for VPC-connected scaling groups.
- The multiple-zone scaling policy of a scaling group cannot be modified.


Auto Scaling cannot create a preemptible instance when the market price of the instance exceeds your bid. This may delay scale-out activities and then affect your service. To avoid this issue, you can configure a cost optimization policy as the multi-zone scaling policy. With this policy, when failing to create a preemptible instance, Auto Scaling attempts to create a pay-as-you-go instance of the same type. This guarantees a high success rate for scaling activities and controls the cost. You can configure a cost optimization policy and multiple instance types at the same time to further increase the chance of successful scaling. Scaling groups that apply cost optimization policies attempt to create ECS instances based on the unit prices of vCPUs, from low to high. Even if you set the billing method of instances to pay-as-you-go, a cost optimization policy guarantees that you can use ECS instance resources at the lowest cost.

## Procedure

1. Create a scaling group. This step describes parameters related to the multi-zone scaling policy. For more information about other parameters of the scaling group, see [Create a scaling group](#).
  - i. Set the network type to VPC and select multiple VSwitches of the VPC. A VSwitch belongs to only one zone. After you configure multiple VSwitches for the scaling group, the scaling group can create ECS instances in multiple zones. This helps you utilize available ECS resources in different zones based on your requirements.
  - ii. Set the multi-zone scaling policy to Cost Optimization.
  - iii. Set other parameters as required.
2. Create a scaling configuration. This step describes parameters related to the multi-zone scaling policy. For more information about other parameters of the scaling configuration, see [创建伸缩配置](#).
  - i. Set the billing method to Preemptible Instance.
  - ii. Select instance types. You can select up to 10 instance types.
    - We recommend that you select instance types with similar performance in terms of the vCPU, memory, physical processor, clock speed, internal network bandwidth, or packet forwarding rate.
    - We recommend that you set a maximum bid for each instance type. If you use automatic bidding, the scaling group bids for and creates preemptible instances at the market price.
    - The specifications of I/O optimized instances vary greatly from those of non-I/O optimized instances. If you choose these two types of instances at the same time, the success rate of scaling activities cannot be increased significantly.
  - iii. Set other parameters as required.
3. Enable the scaling group.
4. Create a scaling rule. This step describes parameters for creating a simple rule to verify the cost optimization policy. For more information about other parameters of the scaling rule, see [Create a scaling rule](#).
  - i. Set the rule type to Simple Scaling Rule.
  - ii. In the Operation field, specify that one ECS instance is added after the scaling rule is triggered.
  - iii. Set other parameters as required.
5. Execute the scaling rule.

## Verification

Assume that in the preceding procedure, you have selected a VSwitch in Zone B of China (Qingdao) and a VSwitch in Zone C of China (Qingdao) for the scaling group, and selected the ecs.sn1.large and ecs.sn1.xlarge instance types for the scaling configuration. The billing method is set to Preemptible Instance. Therefore, instances of a specified type have two unit prices of vCPUs. One is for preemptible instances, and the other is for pay-as-you-go instances.

 **Notice** The prices listed in this topic are used only as examples. Refer to the purchase page of ECS instances for actual prices.



# 10. Use the Alibaba Cloud ESS SDK to create a multi-zone scaling group

This topic describes how to use the Alibaba Cloud ESS SDK to create a multi-zone scaling group in either Java or Python.

## Prerequisites

Before you perform the operations provided in the tutorial, you must have registered an Alibaba Cloud account. To create an Alibaba Cloud account, click [Create a new Alibaba Cloud account](#).

## Context

The network type of a scaling group can be either Virtual Private Cloud (VPC) or classic network. When creating a VPC-connected scaling group, you must configure a VSwitch for the scaling group. After the scaling group is created, all ECS instances that are scaled out through the group use this VSwitch.

Auto Scaling allowed a VPC-connected scaling group to have only one VSwitch configured. A VSwitch belongs to only one zone. If ECS instances cannot be created in the zone where the VSwitch resides due to reasons such as insufficient inventory, the scaling configuration, scaling rules, and monitoring tasks of the scaling group become invalid.

To address the preceding issue and improve the availability of the scaling group, the `VSwitchIds.N` parameter has been added to allow you to create multi-zone scaling groups. When creating a scaling group, you can use the `VSwitchIds.N` parameter to configure multiple VSwitches for the scaling group. When ECS instances cannot be created in the zone where a VSwitch resides, Auto Scaling automatically switches to the zone where a different VSwitch resides. When you use this parameter, note that:

- If the `VSwitchIds.N` parameter is specified, the `VSwitchId` parameter is ignored.
- The `VSwitchIds.N` parameter allows you to specify up to five VSwitches within a VPC in multiple zones when you create a scaling group. Valid values of N are integers from 1 to 5.
- VSwitches specified in the `VSwitchIds.N` parameter must be within the same VPC.
- In the `VSwitchIds.N` parameter, N indicates the priority of each VSwitch. The VSwitch with N set to 1 has the highest priority to create an ECS instance. The greater the N value, the lower the priority.
- When an ECS instance cannot be created in the zone where the VSwitch with the highest priority resides, the instance will be created in the zone where the VSwitch with the second highest priority resides. We recommend that you specify VSwitches in different zones of the same region to avoid failing to create ECS instances due to insufficient inventory in a single zone and improve the availability of scaling groups.

## Use the Alibaba Cloud ESS SDK for Java to create a multi-zone scaling group

1. Install the Alibaba Cloud ESS SDK for Java. Download the `aliyun-java-sdk-core` and `aliyun-java-sdk-ess` dependency libraries. You can visit [Maven Central](#) to search for and download the JAR packages accordingly. The JAR package version for `aliyun-java-sdk-ess` must be V2.1.3 or later and that for `aliyun-java-sdk-core` must be the latest.

You can also use Apache Maven to manage the dependency libraries of your Java projects by adding the following dependencies to the *pom.xml* file.

```
<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-ess</artifactId>
<version>2.1.3</version>
</dependency>
<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-core</artifactId>
<version>3.5.0</version>
</dependency>
```

2. Use the Java SDK to create a multi-zone scaling group. After importing the Alibaba Cloud ESS SDK for Java to a Java project, you can use the SDK code to create a multi-zone scaling group. The sample code is as follows:

```
public class EssSdkDemo {
    public static final String REGION_ID = "cn-hangzhou";
    public static final String AK = "ak";
    public static final String AKS = "aks";
    public static final Integer MAX_SIZE = 10;
    public static final Integer MIN_SIZE = 1;
    public static final String SCALING_GROUP_NAME = "TestScalingGroup";

    // The list of VSwitches. The priority of the VSwitch decreases from the beginning to the end of the
    // list, with the first VSwitch having the highest priority.
    public static final String[] vswitchIdArray = { "vsw-id1", "vsw-id2", "vsw-id3", "vsw-id4", "vsw-id5" };
    public static final List<String> vswitchIds = Arrays.asList(vswitchIdArray);
    public static void main(String[] args) throws Exception {
        IClientProfile clientProfile = DefaultProfile.getProfile(REGION_ID, AK, AKS);
        IAcsClient client = new DefaultAcsClient(clientProfile);
        createScalingGroup(client);
    }

    /**
     * Create a multi-zone scaling group.
     * @param client
     * @return
     * @throws Exception
     */
    public static String createScalingGroup(IAcsClient client) throws Exception {
        CreateScalingGroupRequest request = new CreateScalingGroupRequest();
        request.setRegionId("cn-beijing");
        request.setMaxSize(MAX_SIZE);
        request.setMinSize(MIN_SIZE);
        request.setScalingGroupName(SCALING_GROUP_NAME);
        request.setVSwitchIds(vswitchIds);
        CreateScalingGroupResponse response = client.getAcsResponse(request);
        return response.getScalingGroupId();
    }
}
```

In the preceding code, the priority of the VSwitch decreases from the beginning to the end of the list. The VSwitch at the top of the list has the highest priority.

## Use the Alibaba Cloud ESS SDK for Python to create a multi-zone scaling group

1. Install the Alibaba Cloud ESS SDK for Python. To install the Alibaba Cloud ESS SDK for Python, you must download and install the *aliyun-python-sdk-ess* and *aliyun-python-sdk-core* dependency libraries. We recommend that you use pip to install Python dependency libraries. For more information about how to install pip, visit [Installation-pip](#). After pip is installed, run the `pip install aliyun-python-sdk-ess==2.1.3 pip install aliyun-python-sdk-core==3.5.0` command to install the dependency libraries.
2. Use the Python SDK to create a multi-zone scaling group. After installing the dependency libraries of the Alibaba Cloud ESS SDK for Python, you can use the SDK code to create a multi-zone scaling group. The sample code is as follows:

```
# coding=utf-8
import json
import logging

from aliyunsdkcore import client

from aliyunsdkess.request.v20140828.CreateScalingGroupRequest import CreateScalingGroupRequest

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

# Replace the following ak and aks values with your own AccessKey ID and secret.
ak = 'ak'
aks = 'aks'

scaling_group_name = 'ScalingGroupTest'

max_size = 10

min_size = 1

vswitch_ids = ["vsw-id1", "vsw-id2", "vsw-id3", "vsw-id4", "vsw-id5"]

region_id = 'cn-beijing'

clt = client.AcsClient(ak, aks, region_id)
```

```
def _create_scaling_group():
    request = CreateScalingGroupRequest()
    request.set_ScalingGroupName(scaling_group_name)
    request.set_MaxSize(max_size)
    request.set_MinSize(min_size)
    request.set_VSwitchIds(vswitch_ids)
    response = _send_request(request)
    return response.get('ScalingGroupId')
```

```
def _send_request(request):
    request.set_accept_format('json')
    try:
        response_str = clt.do_action(request)
        logging.info(response_str)
        response_detail = json.loads(response_str)
        return response_detail
    except Exception as e:
        logging.error(e)
if __name__ == '__main__':
    scaling_group_id = _create_scaling_group()
    print 'Scaling group created successfully. Scaling group ID:' + str (scaling_group_id)
```

In the preceding code, the priority of the VSwitch decreases from the beginning to the end of the list. The VSwitch at the top of the list has the highest priority.

## Related information

### Related topics

- [CreateScalingGroup](#)
- [CreateScalingConfiguration](#)
- [Set parameters of scaling configurations to customize ECS instances](#)