Alibaba Cloud
Elastic Compute Service

Elasticity

Issue: 20200619
Legal disclaimer

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

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

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

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

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please contact Alibaba Cloud directly if you discover any errors in this document.
# Document conventions

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal disclaimer</td>
<td>1</td>
</tr>
<tr>
<td>Document conventions</td>
<td>1</td>
</tr>
<tr>
<td>1 Launch template</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Launch templates</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Create a launch template</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Create a template version</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Delete a launch template and a specified template version</td>
<td>6</td>
</tr>
<tr>
<td>2 Create a scaling group based on an ECS instance</td>
<td>8</td>
</tr>
<tr>
<td>3 Deployment sets</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Overview</td>
<td>19</td>
</tr>
<tr>
<td>3.2 Create a deployment set</td>
<td>21</td>
</tr>
<tr>
<td>3.3 Create an ECS instance in a deployment set</td>
<td>22</td>
</tr>
<tr>
<td>3.4 Change the deployment set of an instance</td>
<td>25</td>
</tr>
<tr>
<td>3.5 Manage deployment sets</td>
<td>27</td>
</tr>
<tr>
<td>3.6 Delete a deployment set</td>
<td>29</td>
</tr>
<tr>
<td>4 Manage auto provisioning groups</td>
<td>30</td>
</tr>
<tr>
<td>4.1 Auto provisioning overview</td>
<td>30</td>
</tr>
<tr>
<td>4.2 Configure an auto provisioning group</td>
<td>33</td>
</tr>
<tr>
<td>4.3 Create an auto provisioning group</td>
<td>37</td>
</tr>
<tr>
<td>4.4 View an auto provisioning group</td>
<td>42</td>
</tr>
<tr>
<td>4.5 Modify an auto provisioning group</td>
<td>43</td>
</tr>
<tr>
<td>4.6 Delete auto provisioning groups</td>
<td>44</td>
</tr>
<tr>
<td>5 Terraform</td>
<td>46</td>
</tr>
<tr>
<td>5.1 What is Terraform?</td>
<td>46</td>
</tr>
<tr>
<td>5.2 Install and configure Terraform</td>
<td>47</td>
</tr>
<tr>
<td>5.3 Create an ECS instance</td>
<td>48</td>
</tr>
<tr>
<td>5.4 Create multiple ECS instances</td>
<td>51</td>
</tr>
<tr>
<td>5.5 Deploy a web cluster</td>
<td>54</td>
</tr>
</tbody>
</table>
1 Launch template

1.1 Launch templates

A launch template helps you quickly create an ECS instance. A template contains configurations that you can use to create instances for various scenarios with specific requirements.

A template can include any configurations except passwords. It can include key pairs, RAM roles, instance type, and network configurations.

You can create multiple versions of each template. Each version can contain different configurations. You can then create an instance using any version of the template.

Console operations

- Create a template
- Create multiple versions in one template
- Change the default version
- Use a launch template
- Delete a template or version

API operations

- CreateLaunchTemplate
- CreateLaunchTemplateVersion
- DescribeLaunchTemplates
- DescribeLaunchTemplateVersions
- ModifyLaunchTemplateDefaultVersion
- DeleteLaunchTemplate
- DeleteLaunchTemplateVersion

1.2 Create a launch template

This topic describes how to create a launch template and the precautions you need to take note of when you create a launch template.

Context

Before you create a launch template, take note of the following items:
• You can create up to 30 launch templates in each region.
• When you create a launch template, all parameters are optional. However, if a launch template does not contain required parameters such as the instance type or image, you must specify these parameters when you create an instance by using the launch template.
• You cannot modify a launch template that is created. However, you can create new versions for the launch template.

Create a launch template from the ECS console

You can create launch templates for future use from the ECS console.

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Deployment & Elasticity > Launch Templates.
3. In the top navigation bar, select a region.
4. On the Launch Templates page, click Create Template.

5. On the Launch Template page, complete the basic and advanced configurations.

Note:
The Clone Template section is unavailable if you are creating a launch template for the first time. If you have already created launch templates, you can select an existing launch template and one of its versions, and then modify the configurations.
6. In the **Confirm Configuration** step, enter a template name and a template version description. Click **Create Launch Template**.

![Launch Template](image)

**Note:**
In the **Confirm Configuration** step, confirm the parameters you need for your instances. However, these parameters are optional and can be configured as needed.

7. In the **Created** dialog box that appears, click **View Template** to go to the ECS console and view the launch template that you have created.

You can also use developer tools such as Alibaba Cloud CLI, OpenAPI Explorer, and Alibaba Cloud SDKs to call the **CreateLaunchTemplate** operation to create launch templates.

**Create a launch template on the ECS buy page**

If you want to create an instance and save its configurations for future use, you can create a launch template when you create the instance.

1. Go to the **Elastic Compute Service product homepage**. Click **Buy Now**.
2. On the **Custom Launch** tab that appears, configure the parameters as instructed.
3. In the **Preview** step, click **Save as a Launch Template**.
4. In the dialog box that appears, click **New Template**. Enter a template name and a template version description. Click **Save**.
5. In the **Created** dialog box that appears, click **View Template** to go to the ECS console and view the launch template that you have created.

**Related topics**

#unique_16
1.3 Create a template version

One launch template can have multiple versions. This topic describes how to create a template version and change the default template version.

Limits

- The default version number of a newly created template is 1, and you can create additional versions based on this template.

  ![Note:]
  The version number increments automatically as you create a new version. You cannot customize the version number, but you can set any of the template versions as the default version.

- Each template can have a maximum of 30 versions.
- All parameters are optional when you create a template version. However, if the version that you want to use to create an instance does not have all required parameters (such as the instance type and image), you must specify the required parameters at instance creation.
- A template version cannot be modified after it is created.

Prerequisites

You have already created a launch template.

Create a template version by using the ECS console

To create a template version, follow these steps:

1. In the left-side navigation pane, choose Deployment & Elasticity > Launch Templates.
2. Select a template ID to view its version information, and then click New Version. You can also click New Version in the Actions column.

3. On the Launch Template page, set the parameters.

   **Note:**
   
   You can also go to the Clone Template area, select an existing template and version, and then set the parameters.

4. On the Confirm Configuration page, click New Template Version, and then select a template to save the version.

5. Click Create Launch Template.

6. In the dialog box that appears, click View New Version to view the version you have created.

Create a template version on the ECS purchase page

1. Go to the ECS product details page, and then click Buy Now.

2. On the Custom Launch page, configure the parameters.

3. On the Preview page, click Save as Launch Template.

4. In the dialog box that appears, click New Template Version, and then select a template to save the version.

5. In the Created dialog box, click View New Version to view the version you have created.

Change the default template version

You can set a commonly used template version as the default version to facilitate instance creation. To change the default template version, following these steps:

1. In the left-side navigation pane, choose Deployment & Elasticity > Launch Templates.
2. Select a template ID to view its version information.

3. Find the version you want to set as default, and then click **Set as Default** in the **Actions** column.

You can also complete this task by calling the **ModifyLaunchTemplateDefaultVersion** API action through the Alibaba Cloud CLI, OpenAPI Explorer, or Alibaba Cloud SDK.

1.4 Delete a launch template and a specified template version

This topic describes how to delete a launch template and a specified template version.

### Note:
When you delete a launch template, all versions of the template are also deleted.

**Delete a specified template version**

1. In the left-side navigation pane, choose **Deployment & Elasticity > Launch Templates**.
2. Select the target template ID to view the version information.
3. In the **Version Information** area, find the version you want to delete and, in the **Actions** column, click **Delete**.

### Note:
You cannot delete the default template version. If the version you want to delete is the default version, change it to a non-default version, and then delete it. If you no longer need any versions of a single template, delete the template.

4. Click OK.

You can also complete this task by calling the `DeleteLaunchTemplateVersion` API action through the Alibaba Cloud CLI, OpenAPI Explorer, or Alibaba Cloud SDK.

Delete a launch template

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Deployment & Elasticity > Launch Templates.
3. Find the template you want to delete, and click Delete in the Actions column.
4. Click OK.

You can also complete this task by calling the `DeleteLaunchTemplate` API action through the Alibaba Cloud CLI, OpenAPI Explorer, or Alibaba Cloud SDK.
2 Create a scaling group based on an ECS instance

If your business loads fluctuate frequently or have a fixed scaling pattern, we recommend that you use Auto Scaling to implement automatic scaling. After a scaling group is created based on an ECS instance, Auto Scaling automatically adds or removes a group of ECS instances that use the same scaling configuration based on business needs.

Prerequisites

- Before you associate a scaling group with SLB instances, ensure that the following conditions are met:
- Before you associate a scaling group with ApsaraDB for RDS instances, ensure that the following conditions are met:
  - At least one ApsaraDB for RDS instance is in the Running state. For more information, see #unique_18.
  - The ApsaraDB for RDS instances and the scaling group must reside in the same region.

Context

You can create a scaling group based on an ECS instance regardless of the billing method of the instance. A subscription, pay-as-you-go, or preemptible instance can all be used as a scaling configuration source.

After the scaling group is created, you can manually add existing ECS instances or use Auto Scaling to automatically create ECS instances. Only pay-as-you-go and preemptible ECS instances can be automatically created. However, you can manually add existing ECS instances of any billing method.

For more information about the limits of a scaling group, see #unique_19.

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Instances & Images > Instances.
3. In the top navigation bar, select a region.
4. Find the ECS instance that you want to use as a scaling configuration source, and choose More > Deployment & Elasticity > New auto scaling in the Actions column.

5. On the Create Scaling Group page, view the scaling configuration source.

When you create a scaling group based on an ECS instance, the Select Existing Instance option and the ECS instance are automatically selected in the Source Type section on the Create Scaling Group page. Keep these settings unchanged.

6. Configure basic information of the scaling group.

   a) Specify the scaling group name.

   b) Specify the number of instances in the scaling group.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Number of Instances</td>
<td>When the upper limit of ECS instances is reached, Auto Scaling automatically removes instances to ensure that the number of instances in the scaling group matches the upper limit.</td>
</tr>
<tr>
<td>Minimum Number of Instances</td>
<td>When the lower limit of ECS instances is reached, Auto Scaling automatically adds instances to ensure that the number of instances in the scaling group matches the lower limit.</td>
</tr>
<tr>
<td>Expected Number of Instances</td>
<td>Auto Scaling automatically keeps the number of ECS instances at the expected level. For more information, see #unique_20.</td>
</tr>
</tbody>
</table>

   c) Specify the default cooldown time.

   The Default Cooldown Time parameter specifies the default cooldown time of a scaling group in seconds. During the cooldown time, the scaling group rejects all scaling activity requests triggered by event-triggered tasks from CloudMonitor. However, some scaling activities are not subject to the cooldown time and are
executed immediately. These scaling activities are triggered by other types of tasks such as scheduled tasks and manually executed tasks.

d) Optional: Specify the instance removing policy.

When ECS instances are no longer required, you can specify the policy to remove the instances. If multiple ECS instances meet the conditions of the policy, one instance is removed at random. The Instance Removing Policy parameter contains two fields.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values for the first field</td>
<td><strong>Earliest Instance Created Using Scaling Configuration</strong></td>
<td>The scaling configuration refers to the instance configuration source that contains the scaling configuration and launch template. Auto Scaling filters the instances that use the earliest scaling configuration and launch template. Manually added instances are not associated with a scaling configuration or launch template. Therefore, manually added instances are not filtered first. If all associated instances have been filtered but the scaling group needs to remove more instances, manually added instances are removed at random. The version of a launch template does not necessarily indicate the order in which the template was added. For example, you select the lt-foress V2 template when you create a scaling group. Then, you select the lt-foress V1 template to modify the scaling group. The scaling group considers the lt-foress V2 launch template as the template that was added earlier.</td>
</tr>
<tr>
<td><strong>Earliest Created Instance</strong></td>
<td>Auto Scaling filters instances that were created at the earliest points in time.</td>
<td></td>
</tr>
<tr>
<td><strong>Most Recent Created Instance</strong></td>
<td>Auto Scaling filters instances that were created most recently.</td>
<td></td>
</tr>
</tbody>
</table>
### Elastic Compute Service

#### Elasticity / 2 Create a scaling group based on an ECS instance

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Policy</td>
<td>Auto Scaling does not filter instances.</td>
</tr>
<tr>
<td></td>
<td>Earliest Created Instance</td>
<td>Auto Scaling filters instances that were created at the earliest points in time from the filtering results in the first step.</td>
</tr>
<tr>
<td></td>
<td>Most Recent Created Instance</td>
<td>Auto Scaling filters instances that were created most recently from the filtering results in the first step.</td>
</tr>
</tbody>
</table>

By default, Auto Scaling filters instances that were created based on the earliest scaling configuration. From the filtering results, Auto Scaling filters and removes the instances that were created at the earliest points in time.

e) Optional: Enable deletion protection for the scaling group.

After this feature is enabled, the scaling group cannot be deleted by using the console or API.

f) Add tags.

You can add tags to search for and manage scaling groups. For more information, see #unique_21.

7. Specify the scaling configuration.

a) View the network type.

#### Notice:

Keep the default network type unchanged. Otherwise, the creation fails.

<table>
<thead>
<tr>
<th>Network type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC</td>
<td>• When you create a scaling configuration, you can select only instance types that support VPC.</td>
</tr>
<tr>
<td></td>
<td>• When you manually add existing ECS instances, you can select only instances in the same VPC as the scaling group.</td>
</tr>
</tbody>
</table>
Elastic Compute Service

Elasticity / 2 Create a scaling group based on an ECS instance

<table>
<thead>
<tr>
<th>Network type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Classic Network| • When you create a scaling configuration, you can select only instance types that support the classic network.  
• When you manually add existing ECS instances, you can select only instances in the classic network. |

b) Optional: If the network type is **VPC**, configure the required parameters.

- **VPC**

  **Note:**
  Keep the default VPC unchanged. Otherwise, the creation fails.

- **VSwitch**

  A VSwitch can belong only to one zone. To deploy ECS instances across multiple zones, you must specify multiple VSwitches that belong to different zones. We recommend that you select multiple zones to reduce the risk of insufficient resources and increase the success rate of creating ECS instances.

- **Multi-zone scaling policy**

  **Notice:**
  The multi-zone scaling policy cannot be modified after the scaling group is created.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Priority Policy</strong></td>
<td>The VSwitch that is first selected has a higher priority. When an ECS instance cannot be created in the zone where the VSwitch with the highest priority resides, Auto Scaling automatically uses the VSwitch that has the second highest priority to create the ECS instance.</td>
</tr>
<tr>
<td><strong>Balanced Distribution Policy</strong></td>
<td>The policy takes effect only when the scaling group is associated with multiple VSwitches that are distributed across more than two zones. It evenly distributes ECS instances across zones where the VSwitches reside. If the ECS instances are not evenly distributed across zones due to insufficient resources, you can use the Rebalance Distribution feature to evenly distribute the ECS instances. For more information, see #unique_22.</td>
</tr>
<tr>
<td>Policy</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cost Optimization Policy</td>
<td>The policy takes effect only when you specify multiple instance types in the scaling configuration. Auto Scaling creates ECS instances based on the unit prices of vCPUs in ascending order.</td>
</tr>
<tr>
<td></td>
<td>If you select Preemptible Instance as the billing method in the scaling configuration, preemptible instances are preferentially created. When preemptible instances cannot be created due to insufficient resources, Auto Scaling automatically attempts to create pay-as-you-go instances.</td>
</tr>
</tbody>
</table>

If you select **Cost Optimization Policy**, you can set the parameters listed in the following table to enable the scaling policy that involves both pay-as-you-go and preemptible instances.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Pay-as-you-go Instances</td>
<td>The minimum number of ECS instances that use the pay-as-you-go billing method. Default value: 0. If the number of pay-as-you-go ECS instances in the scaling group is less than this value, Auto Scaling preferentially creates pay-as-you-go instances.</td>
</tr>
<tr>
<td>Percentage of Pay-as-you-go Instances</td>
<td>The percentage of pay-as-you-go ECS instances among all instances that are created automatically. Default value: 70%. The percentage is calculated based on the difference between the total number of instances and the minimum number of pay-as-you-go instances.</td>
</tr>
<tr>
<td>Lowest Cost Instance Types</td>
<td>The number of instance types with the lowest cost. Default value: 1. This parameter takes effect when multiple instance types are specified in the scaling configuration. When preemptible instances are created, Auto Scaling evenly creates ECS instances from the lowest cost instance types.</td>
</tr>
<tr>
<td>Enable Supplemental Preemptible Instances</td>
<td>After the Supplemental Preemptible Instances feature is enabled, Auto Scaling automatically creates new preemptible instances five minutes before the existing instances are reclaimed.</td>
</tr>
</tbody>
</table>

- Instance reclaim mode

**Notice:**
The reclaim mode cannot be modified after the scaling group is created.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Release Mode</strong></td>
<td>When a scale-in event is triggered, Auto Scaling automatically releases a specific number of ECS instances. When a scale-out event is triggered, Auto Scaling automatically creates a specific number of ECS instances for the scaling group.</td>
</tr>
</tbody>
</table>
### Mode Description

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shutdown and Reclaim Mode</strong></td>
<td>This mode can improve scaling efficiency.</td>
</tr>
<tr>
<td></td>
<td>When a scale-in event is triggered, the status of the removed ECS instances becomes No Fees for Stopped Instances (VPC-Connected), and their vCPUs, memory, and public IP addresses are reclaimed. You are no longer billed for these resources. However, you are still billed for other resources such as disks and EIPs. For more information, see #unique_23. These stopped ECS instances form a stopped instance pool.</td>
</tr>
</tbody>
</table>

**Note:**

If the ECS instances have public IP addresses before they enter the No Fees for Stopped Instances (VPC-Connected) state, the instances will be reassigned with public IP addresses when they are restarted, and the addresses may be different from the previous ones.

When a scale-out event is triggered, the ECS instances in the stopped instance pool preferentially enter the running state. If these instances are still insufficient, Auto Scaling will create more ECS instances.

When a scale-out event is triggered, the ECS instances in the stopped instance pool may or may not enter the running state. If the ECS instances in the stopped instance pool cannot enter the running state due to insufficient resources, Auto Scaling will release these instances and create a specific number of ECS instances.

c) Optional: Add existing instances.

If you specify the expected number of instances and then add existing instances, the expected number of instances will automatically increase. For example, when you create a scaling group, you set the expected number of instances to one and then add two existing instances. After the scaling group is created, two existing instances
are added to the scaling group, and the expected number of instances becomes three.

You can enable the scaling group to manage the instance lifecycle.

- If you select **Enable the scaling group to manage the instance lifecycle**, Auto Scaling will automatically release the ECS instances that are manually removed from the scaling group or automatically removed due to being unhealthy.
- If you do not select **Enable the scaling group to manage the instance lifecycle**, the ECS instances that are removed from the scaling group will not be automatically released.


You can associate only a limited number of SLB and ApsaraDB for RDS instances with a scaling group. For more information, see #unique_19.

a) Associate SLB instances.

After you associate SLB instances with the scaling group, ECS instances that are added to the scaling group will be automatically added as SLB backend servers. You can specify a server group for the ECS instances. The following table lists two available server groups.

<table>
<thead>
<tr>
<th>Server group</th>
<th>Port number</th>
<th>Weight</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default server group</td>
<td>You can specify the port number when you configure listeners for SLB instances.</td>
<td>Default value: 50. You can set the weight to other values in the scaling configuration.</td>
<td>The group of ECS instances that are used to receive requests. If the listener is not configured with a VServer group or a primary/secondary server group, requests are forwarded to the ECS instances in the default server group.</td>
</tr>
<tr>
<td>Server group</td>
<td>Port number</td>
<td>Weight</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VServer group</td>
<td>You can specify the port number when you select a VServer group.</td>
<td>Default value: 50. You can set the weight to other values when you select a VServer group.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

If you specify the default server group and multiple VServer groups at the same time, ECS instances are added to all server groups.

b) Associate ApsaraDB for RDS instances.

After the ApsaraDB for RDS instances are specified, the internal IP addresses of ECS instances that are added to the scaling group are added to the whitelists of the ApsaraDB for RDS instances to allow for internal communication.

9. Click **Create Scaling Group**.

10. In the **Create Scaling Group Status Wizard** message, click **View Scaling Groups**.

The newly created scaling group is displayed in the scaling group list. Auto Scaling automatically creates a scaling configuration for the scaling group, and the scaling group enters the **Enabled** state after it is created because an existing ECS instance is used as the scaling configuration source.

- If the value of **Minimum Number of Instances** is greater than zero, Auto Scaling automatically creates ECS instances to ensure that the number of instances in the scaling group is no less than the minimum number of instances.
- If the value of **Expected Number of Instances** is greater than zero, Auto Scaling automatically creates or removes ECS instances to ensure that the number of instances in the scaling group is equal to the expected number of instances.

**What's next**

You can manually or use scheduled or event-triggered tasks to add or remove ECS instances in the scaling group. For more information, see:
• #unique_24
• #unique_25
• #unique_26
• #unique_27

Related topics
#unique_28
3 Deployment sets

3.1 Overview

A deployment set is a distribution policy to control ECS instances and implement disaster recovery and business availability when ECS instances are created.

Deployment policy

You can use a deployment set to distribute your ECS instances to different physical servers to guarantee high availability and set up underlying disaster recovery. When you create ECS instances in a deployment set, Alibaba Cloud will start the ECS instances on different physical servers within the specified region based on the deployment policy that you configured.

Deployment sets support the high availability policy:

- When you use the high availability policy, all the ECS instances within your deployment set are strictly distributed across different physical servers within the specified region. The high availability policy applies to application architectures where several ECS instances must be isolated from each other. The policy significantly reduces the chances of service being unavailable.

- When using the high availability policy, you may not be able to create an ECS instance when there is a supply shortage in the specified region. Furthermore, with the No Fees for Stopped Instances (VPC-Connected) feature enabled, pay-as-you-go instances may fail to start next time. If this problem occurs, we recommend that you wait a while and then try again or create a new instance.

Deployment example

The following figure shows a typical example on how to use a deployment set to improve business reliability. In the deployment set, four ECS instances are distributed to four different physical servers.
If you want to achieve low latency communication between ECS instances, we recommend that you make sure the network types of the instances are the same. For example, select the same VPC for the ECS instances when you create them.

**Billing details**

Deployment sets are free of charge, but you will be charged for the usage of ECS instances, disks, snapshots, images, and public bandwidth in deployment sets. For more information, see #unique_31.

**Limits**

Before you use deployment sets, note that:

- Deployment sets cannot be merged.
- You cannot create preemptable instances in deployment sets.
- You cannot create dedicated hosts in deployment sets.
- When you create ECS instances in a deployment set, you can create up to seven ECS instances in each zone. This limit varies with your ECS usage. You can use the following formula to calculate the number of ECS instances that can be created in an Alibaba Cloud region: \(7 \times \text{Number of zones}\).
- The instance families that can be created in deployment sets include c6, g6, r6, hfc6, hfg6, hfr6, c5, d1, d1ne, g5, hfc5, hfg5, i2, i2g, i1, ic5, r5, se1ne, sn1ne, and sn2ne. For more information about instance types and their performance, see Instance families.
- Supply shortage may result in a failure to create an instance or restart a pay-as-you-go instance that has the No Fees for Stopped Instances (VPC-Connected) feature enabled in a deployment set. For more information, see #unique_23.

For more information about the limits and quotas of deployment sets, see the "Deployment set limits" section in Limits.
3.2 Create a deployment set

This topic describes how to create a deployment set in the ECS console. You can use a deployment set to distribute your ECS instances to different physical servers to achieve high availability and disaster recovery.

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Deployment & Elasticity > Deployment Sets.
3. In the top navigation bar, select a region.
4. On the Deployment Sets page, click Create Deployment Set.
5. In the **Create Deployment Set** dialog box, set **Name** and **Description**.

Currently, the **Strategy** parameter only supports the **High availability** option. For more information about the deployment set strategy, see **Deployment strategy**.

![Create Deployment Set dialog box]

**What's next**

After creating a deployment set, you can perform the following tasks:

- Create an instance in the deployment set

**Related topics**

#unique_38

### 3.3 Create an ECS instance in a deployment set

This topic describes how to create an ECS instance in a deployment set by using the ECS console.

**Prerequisites**

A deployment set is created in the specified region. For more information about how to create a deployment set, see **Create a deployment set**.

**Context**

When you create ECS instances in a deployment set, you can create up to seven ECS instances in each zone. You can use the following formula to calculate the maximum number of ECS instances that you can create in a deployment set within a region: \( 7 \times \text{Number of zones within the region} \). These limits vary with your ECS usage.
When you create an instance, you can use a launch template or use the batch creation feature to facilitate instance creation. For more information, see Launch templates.

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Deployment & Elasticity > Deployment Sets.
3. In the top navigation bar, select a region.
4. On the Deployment Sets page, find the target deployment set.

You can use one of the following methods to create an ECS instance in the deployment set:

- In the Actions column corresponding to the deployment set, click Create Instance.
- Click deployment set ID. In the Instances pane that appears, click Create Instance.
5. On the page that appears, complete the instance configuration on the Custom Launch tab. For more information about how to create an instance, see Create an instance by using the provided wizard.

Note the following configurations when you create an instance:

- **Basic configurations**:
  - **Region**: The ECS instance and the target deployment set must be in the same region.
  - **Zone**: Up to seven ECS instances can be created in each zone of a deployment set.
  - **Instance Type**: The instance families that can be created in deployment sets include c6, g6, r6, hfc6, hfg6, hfr6, c5, d1, d1ne, g5, hfc5, hfg5, i2, i2g, i1, ic5, r5, se1ne, sn1ne, and sn2ne. For more information about instance types and their performance, see Instance families.
  - **Quantity**: optional. You can specify the number of ECS instances to create. This number must be specified based on the number of ECS instances that already exist in the current zone of the deployment set.

- **System Configuration > Sequential Suffix**: optional. After you create multiple instances, you can add sequential suffixes to the instance names and hostnames. The sequential suffix ranges from 001 to 999.

- **Grouping > Deployment Set**: Select the target deployment set.

- **Preview > Save as Launch Template**: optional. You can save your configurations as a launch template that you can use to quickly create an instance the next time. For more information, see Launch templates.

6. Check the settings you have made, and then click Create Order or Create Instance.

7. In the left-side navigation pane, choose Deployment & Elasticity > Deployment Sets. On the Deployment Sets page, find the deployment set to view the instance that you have created.

What's next
After you create an ECS instance, you can perform the following operations:

- View and manage ECS instances in the deployment set. For more information, see instance-related topics.
- Change the deployment set where the ECS instance is located. For more information, see Change the deployment set of an instance.

Related topics
#unique_45

3.4 Change the deployment set of an instance

This topic describes how to change the deployment set of an instance by using the ECS console.

Prerequisites

A deployment set is created for your instance. For more information, see Create a deployment set.

Procedure

1. Find the target instance. The instance must be in the Stopped or Running state.
2. In the **Actions** column, choose **More > Instance Settings > Change Deployment Set**.
3. In the **Change Deployment Set** dialog box, select the destination deployment set and specify the **Force Change** option:

- **Yes**: You can change the host of an instance and restart an instance that is in the **Running** or **Stopped** state.
- **No**: You cannot change the host of an instance. The deployment set must be added on the current host. This may cause the deployment set modification to fail.

4. Click **OK**.

You can also call the **ModifyInstanceDeployment** API action to modify the name and description of a deployment set.

### 3.5 Manage deployment sets

After creating a deployment set, you can modify the deployment set name and description, or remove deployment sets that are no longer required to ensure that the usage limit is not exceeded.

**Edit deployment set information**

To change the name or description of a deployment set in the ECS console, follow these steps:

1. Find the deployment set that needs to be modified.
2. Edit the information using either of the following methods:

- Hover the cursor over the **Deployment Set Name** column, click the icon that appears, and then enter the deployment set name and description.
- In the **Actions** column of the target deployment set, click **Modify Information**, and enter the deployment set name and description.

![Deployment Set Table](image)

3. Click **OK**.

You can also call the **ModifyDeploymentSetAttributes** API operation to modify the deployment set name and description.

**Delete deployment sets**

**Note:**
If a deployment set already includes an instance, you cannot delete the deployment set.

To delete one or more deployment sets in the ECS console, follow these steps:

1. Select one or more deployment sets that need to be deleted, hover the cursor over the **Actions** menu, and then click **Delete**.

![Deployment Set Table](image)

2. Click **OK** to delete the deployment set.

You can use the **DeleteDeploymentSet** API operation to delete deployment sets.
3.6 Delete a deployment set

You can delete any deployment sets that you no longer need to ensure that the usage limit is not exceeded.

Prerequisites

No instances exist in the deployment set. If instances exist in the deployment set, you must release them before you can delete the deployment set.

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Deployment & Elasticity > Deployment Sets.
3. In the top navigation bar, select a region.
4. On the Deployment Sets page, click Delete in the Actions column corresponding to the deployment set that you want to delete.
5. In the message that appears, click OK.

Related topics

#unique_50
4 Manage auto provisioning groups

4.1 Auto provisioning overview

Auto provisioning is a service to quickly deploy an instance cluster comprised of preemptible and pay-as-you-go instances. Auto provisioning supports one-click deployment of instance clusters with specified billing methods, zones, and instance families. Auto provisioning uses auto provisioning groups to schedule and maintain computing resources. You can use auto provisioning groups to provide stable computing power. This alleviates the instability caused by the reclaiming of preemptible instances, and avoids the repeated cumbersome process of manually creating instances.

Introduction to auto provisioning groups

Preemptible instances are low-cost computing resources that are subject to a protection period of one hour. After the protection period expires, preemptible instances and their resources may be reclaimed. You need to pay attention to the availability of preemptible instances in the next hour after the protection period. If they are unavailable, you need to create new ones. As the number of preemptible instances increases, the maintenance time also increases. In this scenario, you can use auto provisioning groups to deploy instance clusters based on a configured target capacity and scheduling policy.

Similar to preemptible instances, auto provisioning groups are applicable to stateless application scenarios such as scalable website services, image rendering, big data analytics, and parallel computing. Auto provisioning groups support flexible policy combinations to alleviate the impact of reclaimed preemptible instances. In addition, the delivery of instance clusters is convenient.

Auto provisioning automatically selects instance types and creates the instance cluster. You do not need to calculate the cost of instances individually. If you select Continuous Delivery and Maintain Capacity, the auto provisioning group automatically compares real-time and target capacities. When preemptible instances are reclaimed, the auto provisioning group selects instance types and then creates new instances to maintain the target capacity that meets business needs at the largest extent and meet your computing power needs at the lowest cost.
Auto provisioning groups can create instance clusters to meet computing power needs based on configured instances types and scheduling policies.

**Features**

- **Cross billing methods**
  
  Preemptible instances provide computing resources at a low cost, but are restricted by the reclaim mechanism and available inventory in a region. Pay-as-you-go instances can be created and released at any time. These instances have a guaranteed inventory, but are much more costly compared to preemptible instances. Auto provisioning groups allow you to create both preemptible instances and pay-as-you-go instances. You can use both billing methods to reduce costs and meet your computing power needs.

- **Cross zones**
  
  Deploying an instance cluster in the same zone reduces network latency between instances, while deploying an instance cluster across zones improves the disaster recovery capabilities of applications. Auto provisioning groups support deployment of instance clusters across zones. You can set zone options as needed.

- **Cross instance families**
  
  Auto provisioning groups allow you to specify alternative instance types across multiple families to provide a range of instance types to select from. Additionally, you can
specify the weight and priority for each instance type to improve task scheduling while ensuring controllability.

- Flexible policy combinations

Auto provisioning groups can meet various dynamic business needs through combinations of target capacities and scale-out policies. Auto provisioning groups allow you to set the target capacities of clusters, preemptible instances, and pay-as-you-go instances, and to specify scaling policies for preemptible instances and pay-as-you-go instances. Additionally, you can specify solutions to fulfill the target capacity of the cluster when preemptible and pay-as-you-go instances are insufficient.

- Complete cost control

Auto provisioning groups allow you to set a maximum price for global and individual instance types to ensure costs remain within your expectations.

- Practical protection mechanism

Auto provisioning groups provide the shutdown option. You can enable this option when an auto provisioning group expires or when instances exceed the target capacity. Routine health checks are performed on instances in an auto provisioning group to ensure instances are available.

**Billing**

Auto provisioning is free to use. However, instance resources created through auto provisioning will incur charges. For more information on billing, see Preemptible instance and Pay-as-you-go.

**Limits**

- Auto provisioning groups cannot schedule resources across regions.
- A maximum of 1,000 instances can be created under each auto provisioning group.
- Only one launch template can be specified for each auto provisioning group. However, you can extend the launch template to implement more configurations. For more information, see Template configurations.
4.2 Configure an auto provisioning group

This topic describes the factors to consider when you configure an auto provisioning group and the process to deploy an instance cluster. In addition, this topic provides configuration solutions for common scenarios.

Procedure to configure an auto provisioning group

You can refer to the following ideas to determine the details of configuring an auto provisioning group. For information about configuration items, see Create an auto provisioning group.

Process to deploy an instance cluster

After an auto provisioning group is started, the instance cluster is automatically deployed based on the group configurations. The deployment process is as follows:

1. The auto provisioning group tries to fulfill the target capacities of preemptible instances and pay-as-you-go instances.
   - Preemptible instances:
     - If Cost Optimization is specified as the scale-out policy, the auto provisioning group selects an instance type with the lowest cost and creates instances of that type. If Instance Types Allowed by Cost Optimization Policy is set, the auto provisioning group selects a specified number of instance types with the lowest cost and creates instances of those types. For example, if Instance Types Allowed
by Cost Optimization Policy is set to 2, the auto provisioning group selects the two instance types with the lowest cost and creates instances of those types.

- If Distribution Balancing is specified as the scale-out policy, the auto provisioning group creates instances of the selected types and then distributes them evenly among specified zones.

**Note:**

Preemptible instances are reclaimed based on instance types, and instance resources in the same instance family are shared. If you select Distribution Balancing, we recommend that you configure different instance families to avoid all instances being reclaimed at the same time and ensure the cluster remains highly available.

- Pay-as-you-go instances:
  - If Cost Optimization is specified as the scaling-out policy, the auto provisioning group selects an instance type with the lowest cost and creates instances of that type.
  - If Priority-based is specified as the scale-out policy, the auto provisioning group creates instances based on the configured priorities of the instance types.

2. If the specified target capacities of preemptible and pay-as-you-go instances do not meet the target capacity requirements of the cluster, the auto provisioning group creates new instances based on Default Billing Method of Supplemental Instances to meet the capacity requirements.

3. If Continuous Delivery and Maintain Capacity is selected, the auto provisioning group continuously compares real-time and target capacities. If any of the target capacities has not been met, the auto provisioning group creates instances when resources are available to meet the target capacity.

**Example of machine learning scenarios**

Assume that you plan to complete a machine learning task in the next week. The task is used to analyze risk factors for mortgage loans. You have the following requirements for the instance cluster:

- The minimum computing power of a single node is 8 vCPUs and 60 GiB.
- The target computing power of the cluster must be 10 times the minimum computing power of a single node.
To reduce costs, only preemptible instances can be used. It is acceptable if the instance cluster capacity does not reach the target capacity.

Instances must be released after the task is completed.

Considering the preceding requirements, the following configurations are used.

The following two solutions can meet capacity requirements:

- Ten ecs.gn5-c8g1.2xlarge instances
- Five ecs.gn5-c8g1.4xlarge instances

When using the Cost Optimization policy to create preemptible instances, the auto provisioning group compares the required costs of each solution for the instance cluster. The solution with the lowest cost is then selected to implement a one-time delivery to the cluster. If the actual instance cluster capacity does not reach the target capacity, the auto provisioning group does not create new instances again.

### Example of ticketing website scenarios

Assume that you need to build a ticketing website to provide reliable ticketing services at all hours, especially during peak hours, and have the following requirements for the instance cluster:

- The minimum computing power of a single node is 8 vCPUs and 16 GiB.
- The target computing power of the cluster must be 30 times the minimum computing power of a single node.
- The minimum computing power of the cluster must be 20 times the minimum computing power of a single node.
- The website access experience is optimized based on the minimum computing requirements of the cluster to minimize costs.
- The cluster must have disaster recover capabilities.
Considering the preceding requirements, the following configurations are used.

The Distribution Balancing policy is used to create preemptible instances. To meet the requirements for distribution balancing, the auto provisioning group creates instances in each zone. Additionally, the computing power of the instances created by the auto provisioning group must meet the overall computing power requirements. The following combination is used as an example:

- One `ecs.c5.2xlarge` instance, two `ecs.c5.4xlarge` instances, one `ecs.sn1ne.2xlarge` instance, and two `ecs.sn1ne.4xlarge` instances
- Three `ecs.c5.2xlarge` instances, one `ecs.c5.4xlarge` instance, three `ecs.sn1ne.2xlarge` instances, and one `ecs.sn1ne.4xlarge` instance

The Cost Optimization policy is used to create pay-as-you-go instances. The following available solutions meet capacity requirements:

- Twenty `ecs.c5.2xlarge` instances
- Ten `ecs.c5.4xlarge` instances
- Twenty `ecs.sn1ne.2xlarge` instances
- Ten `ecs.sn1ne.4xlarge` instances

The auto provisioning group compares the costs required to deliver pay-as-you-go instances using each solution, and uses the lowest-cost solution to deliver the instance cluster.

In Continuous Delivery and Maintain Capacity mode, the auto provisioning group continuously compares the real-time capacities and target capacities. If preemptible instances fail to be created or are reclaimed, the auto provisioning group creates instances to meet the capacity requirements when resources are available.
4.3 Create an auto provisioning group

This topic describes how to create an auto provisioning group in the ECS console. The created group will deploy an instance cluster as configured.

Prerequisites

- Make sure the user account is granted permissions on auto provisioning. When you access the auto provisioning page for the first time, follow the instructions to assign the user account the RAM role of AliyunECSAutoProvisioningGroupRole.
- A launch template is already created. For more information, see Create a launch template.

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Deployment & Elasticity > Auto Provisioning.
3. In the top navigation bar, select a region.
4. Click Create Auto Provisioning Group.
   To learn about the configurations of an auto provisioning group, see Procedure to configure an auto provisioning group.
5. Configure the auto provisioning group.
   a) Configure the basic information.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Name</td>
<td>The name of the auto provisioning group. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https:. It can contain letters, digits, colons (:), underscores (_), and hyphens (-).</td>
</tr>
</tbody>
</table>
| Group Type      | - **One-time Delivery**: After being started, the auto provisioning group attempts to create an instance cluster with a specified capacity for only one time. If the creation fails, the group does not attempt again.  
                   - **Continuous Delivery and Maintain Capacity**: After being started, the auto provisioning group continuously attempts to create an instance cluster of a specified capacity. The auto provisioning group continuously compares the real-time capacity and target capacity. If there is a gap, the group will attempt to create instances to meet the target capacity. |
### Effective Time

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Time</td>
<td>The time when the auto provisioning group is started. The period of time between this point in time and the point in time specified by Expiration Time is the effective time period of the auto provisioning group.</td>
</tr>
<tr>
<td></td>
<td>• Now: The auto provisioning group immediately executes scheduling tasks after it is created.</td>
</tr>
<tr>
<td></td>
<td>• The scheduled time at which the auto provisioning group takes effect and executes scheduling tasks.</td>
</tr>
</tbody>
</table>

### Expiration Time

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expiration Time</td>
<td>The time when the auto provisioning group expires. The period of time between this point in time and the point in time specified by Effective Time is the effective time period of the auto provisioning group.</td>
</tr>
<tr>
<td></td>
<td>• Never: The auto provisioning group will never expire.</td>
</tr>
<tr>
<td></td>
<td>• The scheduled time at which the auto provisioning group expires.</td>
</tr>
</tbody>
</table>

b) Specify the capacity of the auto provisioning group.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Capacity</td>
<td>The total capacity of the instance cluster, which consists of the following three parts:</td>
</tr>
<tr>
<td></td>
<td>• The capacity of preemptible instances.</td>
</tr>
<tr>
<td></td>
<td>• The capacity of pay-as-you-go instances.</td>
</tr>
<tr>
<td></td>
<td>• The supplemental capacity. The Default Billing Method of Supplemental Instances parameter determines the type of instances to supplement when fulfilling the target capacity.</td>
</tr>
<tr>
<td>Target Capacity of Preemptible Instances</td>
<td>The part of Target Capacity that preemptible instances account for.</td>
</tr>
<tr>
<td>Target Capacity of Pay-As-You-Go Instances</td>
<td>The part of Target Capacity that pay-as-you-go instances account for.</td>
</tr>
</tbody>
</table>

6. Click Next: Template Configurations.

7. Configure the template parameters.

The template configurations include specifying a launch template and making extended configurations that are used to determine alternative instance types. The launch template and extended configurations are associated with the auto provisioning group.
After the auto provisioning group is started, the instance types created are determined by **Capacity-related Settings**.

a) Select a launch template.

Specify a launch template and its version. For more information, see **Launch templates**.

You can also click **View Template** to view the launch template information.

b) Make the extended configurations.

The launch template only specifies one instance type. You can specify more alternative instance types through the extended configurations. To improve the success rate of scheduling, you must specify at least one extended launch template.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Zone</strong></td>
<td>A zone different from that specified in the launch template. This parameter and the <strong>VSwitch</strong> parameter determine the location of the alternative instance types.</td>
</tr>
<tr>
<td><strong>VSwitch</strong></td>
<td>A VSwitch different from that specified in the launch template. This parameter and the <strong>Zone</strong> parameter determine the location of the alternative instance types.</td>
</tr>
<tr>
<td><strong>Instance Type</strong></td>
<td>You can select instance types that share the same instance type family, vCPU, or memory. This allows for more flexibility in scheduling by providing multiple alternative instance types that can be applied to the same scenario.</td>
</tr>
<tr>
<td><strong>Maximum Price</strong></td>
<td>The maximum price for created instances with specified types. If the specified maximum price of an instance types is different from the global maximum price, the maximum price is the lower value out of the two.</td>
</tr>
<tr>
<td><strong>Capacity Weight</strong></td>
<td>The weight is calculated based on the computing power of a specified instance type and the minimum computing power of a single node of the cluster. A greater weight indicates that the instance has more computing power, and as a result fewer instances are required. For example, when the minimum computing power of a single node is 8 vCPUs and 60 GiB of memory, the weight of the instance type with 8 vCPUs and 60 GiB of memory is 1, and the weight of the instance type with 16 vCPUs and 120 GiB of memory is 2.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Priority</td>
<td>This parameter is used together with the <strong>Priority-based</strong> policy for creating pay-as-you-go instances. A value of 0 indicates the highest priority. As the value of this parameter increases, the priority decreases. When creating pay-as-you-go instances, the auto provisioning group selects the instance types in descending order of priority.</td>
</tr>
</tbody>
</table>

c) Click **Save**.


d) Configure the capacity-related settings.

**Capacity-related Settings** is a set of scheduling policies that the auto provisioning group uses to execute scheduling tasks and deploy the instance cluster with a specified capacity. The instance types of the cluster are specified in **Template configurations**.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Billing Method of Supplemental Instances</td>
<td>If the total target capacity of preemptible instances and Pay-As-You-Go instances is less than the capacity of the instance cluster, the auto provisioning group will create instances of the specified type to meet the capacity requirements.</td>
</tr>
</tbody>
</table>
| Policy for Creating Pay-As-You-Go Instances | The policy followed when you create Pay-As-You-Go instances. Options:
  - **Cost Optimization**: Select the instance type of the lowest cost from the alternative instance types.
  - **Priority-based**: Create instances based on the specified priority. When this option is selected, you must specify the priority in the **Extended Configurations** area in the next step. For more information, see **Template configurations**. |

**Note**: If you use a reserved instance to match instances, we recommend that you select the **Priority-based** policy and set a higher priority for the types of matching instances. After the instances are matched, the reserved instance discount is automatically applied to the created Pay-As-You-Go instances. For more information, see **#unique_58**.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Policy for Creating Preemptible Instances</strong></td>
<td>The scale-out policy followed when you create preemptible instances. Options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Cost Optimization</strong>: Select the instance type of the lowest cost from the alternative instance types. When this option is selected, you can set <strong>Instance Types Allowed by Cost Optimization Policy</strong> to specify the allowed number of instance types. The instance types of the lowest cost are used.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Distribution Balancing</strong>: Select this option to evenly create instances across zones where you can purchase alternative instance types.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Preemptible instances are reclaimed based on instance types, and instance resources in the same instance family are shared. If you select Distribution Balancing, we recommend that you configure different instance families to avoid all instances being reclaimed at the same time and ensure the cluster remains highly available.</td>
</tr>
<tr>
<td><strong>Instance Types Allowed by Cost Optimization Policy</strong></td>
<td>Used together with the <strong>Cost Optimization</strong> policy for creating preemptible instances, the auto provisioning group will create preemptible instances by using the specified number of instance types of the lowest cost.</td>
</tr>
<tr>
<td><strong>Shutdown Policy for Preemptible Instances</strong></td>
<td>The action to be performed on preemptible instances when the auto provisioning group expires or exceeds the target capacity. This parameter must be used together with the <strong>Shut Down Instances Upon Group Expiration</strong> and <strong>Shut Down Excess Instances When Target Capacity Is Exceeded</strong> parameters. Options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Stop</strong>: Preemptible instances are stopped and their data is retained.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Release</strong>: Preemptible instances are released and their data is not retained. Released data cannot be recovered.</td>
</tr>
<tr>
<td><strong>Global Maximum Price for Preemptible Instances</strong></td>
<td>The maximum price of created preemptible instances. This parameter applies to all instance types. If the specified maximum price of an instance type is different from the global maximum price, the maximum price is the lower value out of the two.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Shut Down Instances Upon Group Expiration</td>
<td>The policy followed when the auto provisioning group expires. When this option is selected, the auto provisioning group will stop or release the instances based on <strong>Shutdown Policy for Preemptible Instances</strong> after expiration.</td>
</tr>
<tr>
<td>Shut Down Excess Instances When Target Capacity Is Exceeded</td>
<td>The policy followed when the capacity of instances in the auto provisioning group exceeds the target capacity. When this option is selected, the auto provisioning group will stop or release the instances based on <strong>Shutdown Policy for Preemptible Instances</strong> when the target capacity is exceeded.</td>
</tr>
</tbody>
</table>

8. Click **Next: Preview**.

9. Confirm the configurations of the auto provisioning group and template, and click **Create**.

Result

After an auto provisioning group is created, it will start and attempt to deliver the instance cluster at the specified time. If **Continuous Delivery and Maintain Capacity** is selected, the auto provisioning group will continuously maintain the instance cluster. When preemptible instances are reclaimed, the auto provisioning group will attempt to create new instances to meet the target capacity and replace unhealthy instances.

4.4 View an auto provisioning group

This topic describes how to view the information about an auto provisioning group, including its instance information and scheduling task execution.

Procedure

1. Log on to the **ECS console**.
2. In the left-side navigation pane, choose **Deployment & Elasticity > Auto Provisioning**.
3. In the top navigation bar, select a region.
4. Click the ID of an auto provisioning group.

The following table describes details of the auto provisioning group:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Configurations</td>
<td>Includes the basic information and capacity-related settings of the auto provisioning group. For more information, see <strong>Group configurations</strong>.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Template Configurations</td>
<td>Describes the template configurations that determine the alternative instance types available to the auto provisioning group. For more information, see Template configurations.</td>
</tr>
<tr>
<td>Instances</td>
<td>Lists information about instances within the auto provisioning group.</td>
</tr>
<tr>
<td>Group History</td>
<td>Lists the records of scheduling tasks in the auto provisioning group. You can view the results of instance creation tasks in the Task Details column. If most of your scheduling tasks are in the Failed state, you must check your configurations to ensure there is no conflict with actual resource or price requirements. For example, the configured alternative instance types may be limited or the maximum price may be too low.</td>
</tr>
</tbody>
</table>

### 4.5 Modify an auto provisioning group

This topic describes how to modify the configurations of an auto provisioning group.

**Procedure**

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Deployment & Elasticity > Auto Provisioning.
3. In the top navigation bar, select a region.
4. Locate the auto provisioning group that you want to modify, and click **Modify** in the Actions column.
5. Modify the parameters, and click **OK**.

The following parameters can be modified. For more information, see Create an auto provisioning group.

<table>
<thead>
<tr>
<th>Category</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Information</td>
<td>Group Name</td>
</tr>
<tr>
<td>Category</td>
<td>Parameter</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Group Capacity</td>
<td>• Target Capacity</td>
</tr>
<tr>
<td></td>
<td>• Target Capacity of Preemptible Instances</td>
</tr>
<tr>
<td></td>
<td>• Target Capacity of Pay-As-You-Go Instances</td>
</tr>
</tbody>
</table>

**Note:**
Modifying the capacity of an auto provisioning group will affect the scheduling task. After modification, the auto provisioning group, of the One-time Delivery type or the Continuous Delivery and Maintain Capacity type, will perform the scheduling task.

<table>
<thead>
<tr>
<th>Category</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity-related Settings</td>
<td>• Default Billing Method of Supplemental Instances</td>
</tr>
<tr>
<td></td>
<td>• Global Maximum Price for Preemptible Instances</td>
</tr>
<tr>
<td></td>
<td>• Shut Down Instances Upon Group Expiration</td>
</tr>
<tr>
<td></td>
<td>• Shut Down Excess Instances When Target Capacity Is Exceeded</td>
</tr>
</tbody>
</table>

**Note:**
Modifying the capacity-related settings will affect the scheduling task. After modification, the auto provisioning group, of the One-time Delivery type or the Continuous Delivery and Maintain Capacity type, will perform the scheduling task.

### 4.6 Delete auto provisioning groups

This topic describes how to delete auto provisioning groups.

**Procedure**

1. Log on to the ECS console.
2. In the left-side navigation pane, choose Deployment & Elasticity > Auto Provisioning.
3. In the top navigation bar, select a region.

4. Locate the auto provisioning group that you want to delete, and click **Delete** in the **Actions** column.

   You can also select multiple auto provisioning groups and click **Delete Group** to batch delete the groups.

5. Choose whether to delete the instances in the auto provisioning group.

   An auto provisioning group can mitigate the effects caused by reclaimed preemptible instances. If you choose not to delete the instances, you must monitor the running status of the preemptible instances to avoid service interruptions.

6. Click **OK**.
5 Terraform

5.1 What is Terraform?

Terraform is an open source tool for securely and efficiently provisioning and managing cloud infrastructure.

HashiCorp Terraform is an automated IT infrastructure orchestration tool that can use codes to manage and maintain IT resources. The Command Line Interface (CLI) of Terraform provides a simple mechanism, which is used for deploying and versioning configuration files on Alibaba Cloud or any other supported cloud.

Terraform writes the infrastructure, for example, virtual machines, storage accounts, and network interfaces in the configuration file that describes the cloud resource topology. The Command Line Interface (CLI) of Terraform provides a simple mechanism, which is used for deploying and versioning configuration files on Alibaba Cloud or any other supported cloud.

Terraform is a highly scalable tool that supports new infrastructure through providers. You can use Terraform to create, modify, or delete multiple resources, such as ECS, VPC, RDS, and SLB.

Benefits

- **Multiple-cloud infrastructure deployment**

  Terraform applies to multi-cloud scenarios, where similar infrastructure is deployed on Alibaba Cloud, other cloud providers, or local data centers. Developers can use the same tool and configuration file to simultaneously manage the resources of different cloud providers.

- **Automated infrastructure management**

  Terraform can create configuration file templates to define, provision, and configure ECS resources in a repeatable and predictable manner, reducing deployment and management errors resulting from human intervention. In addition, Terraform can deploy the same template multiple times to create the same development, test, and production environment.
• **Infrastructure as code**

With Terraform, you can use codes to manage and maintain resources. It allows you to store the infrastructure status, so that you can track the changes in different components of the system (infrastructure as code) and share these configurations with others.

• **Reduced development costs**

You can reduce costs by creating on-demand development and deployment environments, In addition, you can evaluate such environments before making system changes.

**Application scenarios**

Terraform is a well-proven and open-source automated operation and maintenance tool for managing cloud infrastructure, creating images, and supporting multi-cloud business scenarios.

For the application scenarios of Terraform, see Terraform details.

**Use Terraform**

Terraform allows you to use a simple template language to easily define, preview, and deploy cloud infrastructure on Alibaba Cloud. The steps for Terraform to provision resources in ECS are described as follows:

1. Install Terraform.
2. Configure Terraform.
3. Use Terraform to create one or more ECS instances.

**More information**

• Terraform Alibaba provider
• Terrafrom Alibaba github
• Terraform Registry Alibaba Modules

**5.2 Install and configure Terraform**

You must install and configure Terraform before you can use its simple template language to define, preview, and deploy cloud infrastructure.

**Procedure**

1. Download the appropriate software package for your operating system from the official website of Terraform.
2. Decompress the package to `/usr/local/bin`.
   If you decompress the executable file to another directory, you must define a global path for the file by using one of the following methods:
   - For Linux operating systems, follow the method as described in How to define a global path on Linux.
   - For Windows operating systems, follow the method as described in How to define a global path on Windows.
   - For macOS operating systems, follow the method as described in How to define a global path on macOS.

3. Run the `terraform` command to verify the path.
   If the following similar list of available Terraform options is displayed, the installation is complete:
   ```
   username:~$ terraform
   ```

4. Create and authorize a RAM user to improve the flexibility and security of permission management.
   a. Log on to the RAM console.
   b. Create a RAM user named Terraform and create an AccessKey pair for the user. For more information, see #unique_65.
   c. Authorize the RAM user. In this example, the AliyunECSFullAccess and AliyunVPCFullAccess permissions are granted to the Terraform user. For more information, see #unique_66.

5. Create an environment variable to store authentication information.
   ```
   export ALICLOUD_ACCESS_KEY="LTAIUrZCw3********"
   export ALICLOUD_SECRET_KEY="zfwwWAMWIAiooj14GQ2*************"
   export ALICLOUD_REGION="cn-beijing"
   ```

5.3 Create an ECS instance

This topic describes how to create an Elastic Compute Service (ECS) instance by using Terraform.

Procedure
1. Create a VPC and a VSwitch.

Terraform 0.11 is used in this example.

**Note:**

In Terraform 0.11 and earlier versions, the example usage of the variable expression is `vpc_id = "${alicloud_vpc.vpc.id}"`. In Terraform 0.12 and later versions, the example usage of the variable expression is updated to `vpc_id = "alicloud_vpc.vpc.id"`. Use the corresponding variable expression based on your Terraform version.

a) Create the `terraform.tf` file, enter the following content, and then save the file to the current working directory.

```terraform
resource "alicloud_vpc" "vpc" {
  name       = "tf_test_foo"
  cidr_block = "172.16.0.0/12"
}

resource "alicloud_vswitch" "vsw" {
  vpc_id            = "${alicloud_vpc.vpc.id}"
  cidr_block        = "172.16.0.0/21"
  availability_zone = "cn-beijing-b"
}
```

b) Run the `terraform init` command to initialize the environment.

c) Run the `terraform plan` command to view resources.

d) After you confirm that the resources are correct, run the `terraform apply` command to create the VPC and VSwitch.

e) Run the `terraform show` command to view the created VPC and VSwitch.

You can also log on to the VPC console to view the attributes of the VPC and VSwitch.

2. Create a security group and apply the security group to the created VPC.

a) In the `terraform.tf` file, add the following content:

```terraform
resource "alicloud_security_group" "default" {
  name = "default"
  vpc_id = "${alicloud_vpc.vpc.id}"
}

resource "alicloud_security_group_rule" "allow_all_tcp" {
  type      = "ingress"
  ip_protocol = "tcp"
  nic_type    = "intranet"
  policy      = "accept"
  port_range  = "22/22"
  priority    = 1
  security_group_id = "${alicloud_security_group.default.id}"
  cidr_ip     = "0.0.0.0/0"
}
```

Issue: 20200619
b) Run the terraform plan command to view resources.

3. Create an ECS instance.
   a) In the terraform.tf file, add the following content:

```
resource "alicloud_instance" "instance" {
  # cn-beijing
  availability_zone = "cn-beijing-b"
  security_groups = ["${alicloud_security_group.default.*.id}"

  # series III
  instance_type        = "ecs.n2.small"
  system_disk_category = "cloud_efficiency"
  image_id             = "ubuntu_140405_64_40G_cloudinit_20161115.vhd"
  instance_name        = "test_foo"
  vswitch_id           = "${alicloud_vswitch.vsw.id}"
  internet_max_bandwidth_out = 10
  password             = "<replace_with_your_password>"
}
```

Note:
- In the preceding example, internet_max_bandwidth_out is set to 10, which will cause the ECS instance to be automatically assigned a public IP address.
- For more information about the parameters, visit the Argument Reference section in alicloud_instance.

b) Run the terraform plan command to view resources.

c) After you confirm that the resources are correct, run the terraform apply command to create the ECS instance.

d) Run the terraform show command to view the created ECS instance.

e) Run the ssh root@<publicip> command and enter the password to access the ECS instance.
5.4 Create multiple ECS instances

This topic describes how to create multiple Elastic Compute Service (ECS) instances at a time by using Terraform.

Procedure

1. Create a VPC and a VSwitch.

Terraform 0.11 is used in this example.

---

**Note:**

In Terraform 0.11 and earlier, the example usage of the variable expression is `vpc_id = "${alicloud_vpc.vpc.id}"`. In Terraform 0.12 and later, the example usage of the variable expression is `vpc_id = "${alicloud_vpc.vpc.id}"`. Now, it's `vpc_id = "${alicloud_vpc.vpc.id}"`. The difference is in the way variables are used.
expression is updated to vpc_id = "alicloud_vpc.vpc.id". Use the corresponding variable expression based on your Terraform version.

a) Create the `terraform.tf` file, enter the following content, and then save the file to the current working directory.

```terraform
resource "alicloud_vpc" "vpc" {
  name       = "tf_test_foo"
  cidr_block = "172.16.0.0/12"
}

resource "alicloud_vswitch" "vsw" {
  vpc_id            = "${alicloud_vpc.vpc.id}" 
  cidr_block        = "172.16.0.0/21"
  availability_zone = "cn-beijing-b"
}
```

b) Run the `terraform init` command to initialize the environment.

c) Run the `terraform plan` command to view resources.

d) After you confirm that the resources are correct, run the `terraform apply` command to create the VPC and VSwitch.

e) Run the `terraform show` command to view the created VPC and VSwitch.

You can also log on to the VPC console to view the attributes of the VPC and VSwitch.

2. Create a security group and apply the security group to the created VPC.

a) In the `terraform.tf` file, add the following content:

```terraform
resource "alicloud_security_group" "default" {
  name = "default"
  vpc_id = "${alicloud_vpc.vpc.id}"
}

resource "alicloud_security_group_rule" "allow_all_tcp" {
  type            = "ingress"
  ip_protocol     = "tcp"
  nic_type        = "internet"
  policy          = "accept"
  port_range      = "22/22"
  priority        = 1
  security_group_id = "${alicloud_security_group.default.id}"
  cidr_ip         = "0.0.0.0/0"
```

b) Run the `terraform apply` command to create the VPC, VSwitch, and security group.
b) Run the `terraform plan` command to view resources.

c) After you confirm that the resources are correct, run the `terraform apply` command to create the VPC and VSwitch.

d) Run the `terraform show` command to view the created security group and added security group rule.

You can also log on to the ECS console to view the security group and security group rule.

3. Use a module to create multiple ECS instances. In this example, three ECS instances are created.

a) In the `terraform.tf` file, add the following content:

```terraform
module "tf-instances" {
  source = "alibaba/ecs-instance/alicloud"
  vswitch_id = "{alicloud_vswitch.vsw.id}"  
  group_ids = ["{alicloud_security_group.default. *.id}" ]
  availability_zone = "cn-beijing-b"
  disk_category = "cloud_ssd"
  disk_name = "my_module_disk"
  disk_size = "50" 
  number_of_disks = 7
  instance_name = "my_module_instances_"
  host_name = "sample"
  internet_charge_type = "PayByTraffic"
  number_of_instances = "3"
  password="User@123"
}
```

**Note:**

- In the preceding example, `internet_max_bandwidth_out` is set to 10, which will cause the ECS instances to be automatically assigned public IP addresses.
- For more information about the parameters, visit `ecs-instance`.

b) Run the `terraform plan` command to view resources.

c) After you confirm that the resources are correct, run the `terraform apply` command to create the VPC and VSwitch.

d) Run the `terraform show` command to view the created ECS instances.

e) Run the `ssh root@<publicip>` command and enter the password to access the ECS instances.

```terraform
provider "alicloud" {}
```
5.5 Deploy a web cluster

When you deploy a website or an application, you must deploy multiple nodes. The number of nodes can be scaled up or down automatically based on the number of visits or resource usage amount. Server Load Balancer (SLB) can distribute requests to these nodes dynamically. This topic describes how to deploy a web cluster by using Terraform.

Context

In this example, the entire application is deployed in a single zone and allows access to the Hello World page only through port 8080.
Procedure

1. Create a VPC and a VSwitch.

Terraform 0.11 is used in this example.

**Note:**

In Terraform 0.11 and earlier, the example usage of the variable expression is `vpc_id = "${{alicloud_vpc.vpc.id}}"`. In Terraform 0.12 and later, the example usage of the variable expression is updated to `vpc_id = "alicloud_vpc.vpc.id"`. Use the corresponding variable expression based on your Terraform version.

a) Create the terraform.tf file, enter the following content, and then save the file to the current working directory.

```terraform
resource "alicloud_vpc" "vpc" {  
  name       = "tf_test_foo"  
  cidr_block = "172.16.0.0/12"  
}

resource "alicloud_vswitch" "vsw" {  
  vpc_id            = "${{alicloud_vpc.vpc.id}}"  
  cidr_block        = "172.16.0.0/21"  
  availability_zone = "cn-beijing-b"  
}
```

b) Run the `terraform init` command to initialize the environment.

c) Run the `terraform plan` command to view resources.

d) After you confirm that the resources are correct, run the `terraform apply` command to create the VPC and VSwitch.

e) Run the `terraform show` command to view the created VPC and VSwitch.

You can also log on to the VPC console to view the attributes of the VPC and VSwitch.

2. Create a security group and apply the security group to the created VPC.

a) In the terraform.tf file, add the following content:

```terraform
resource "alicloud_security_group" "default" {  
  name = "default"  
  vpc_id = "${{alicloud_vpc.vpc.id}}"  
}

resource "alicloud_security_group_rule" "allow_all_tcp" {  
  type = "ingress"  
  ip_protocol = "tcp"  
  nic_type = "internet"  
  policy = "accept"  
  port_range = "1/65535"  
  priority = 1  
  security_group_id = "${{alicloud_security_group.default.id}}"  
  cidr_ip = "0.0.0.0/0"  
}
```

Issue: 20200619
b) Run the `terraform plan` command to view resources.

c) After you confirm that the resources are correct, run the `terraform apply` command to create the VPC and VSwitch.

d) Run the `terraform show` command to view the created security group and added security group rule.

You can also log on to the ECS console to view the security group and security group rule.

3. Create an SLB instance and assign a public IP address to it. In this example, a mapping from frontend port 80 to backend port 8080 is configured for the SLB instance and the public IP address of the SLB instance is displayed for subsequent tests.

   a) Create the `slb.tf` file and add the following content:

   ```
   resource "alicloud_slb" "slb" {
     name       = "test-slb-tf"
     vswitch_id = "${alicloud_vswitch.vsw.id}"
     internet   = true
   }
   resource "alicloud_slb_listener" "http" {
     load_balancer_id = "${alicloud_slb.slb.id}"
     backend_port    = 8080
     frontend_port   = 80
     bandwidth       = 10
     protocol        = "http"
     sticky_session  = "on"
     sticky_session_type = "insert"
     cookie          = "testslblistenercookie"
     cookie_timeout  = 86400
     health_check    = "on"
     health_check_type = "http"
     health_check_connect_port = 8080
   }

   output "slb_public_ip"{
     value = "${alicloud_slb.slb.address}"
   }
   ```

   b) Run the `terraform plan` command to view resources.

c) After you confirm that the resources are correct, run the `terraform apply` command to create the VPC and VSwitch.

d) Run the `terraform show` command to view the created SLB instance.

You can also log on to the SLB console to view the created SLB instance.
4. Create Auto Scaling resources.

In this example, the following resources are created:

- **Scaling group**: Specify 2 as the minimum number of instances and 10 as the maximum number of instances in the template, and attach the created SLB instance to the scaling group. Because scaling groups depend on SLB listener configurations, you must use the `depends_on` attribute to specify the deployment sequence in the template.

- **Scaling group configuration**: Specify the specific configuration of the ECS instance in the template. The initialization configuration (user-data) generates a Hello World page and provides services over port 8080. To simplify operations, this example assigns a public IP address to the virtual machine and sets `force_delete` to true to subsequently delete the environment.

- **Scaling rule**: Define the specific scaling rule.

a) Create the `ess.tf` file and add the following content:

```terraform
resource "alicloud_ess_scaling_group" "scaling" {
    min_size = 2
    max_size = 10
    scaling_group_name = "tf-scaling"
    vswitch_ids = ["${alicloud_vswitch.vsw. * .id}"
    loadbalancer_ids = ["${alicloud_slb.slb. * .id}"]
    removal_policies = ["OldestInstance", "NewestInstance"]
    depends_on = ["alicloud_slb_listener.http"]
}

resource "alicloud_ess_scaling_configuration" "config" {
    scaling_group_id = "${alicloud_ess_scaling_group.scaling.id}"
    image_id = "ubuntu_140405_64_40G_cloudinit_20161115.vhd"
    instance_type = "ecs.n2.small"
    security_group_id = "${alicloud_security_group.default.id}"
    active=true
    enable=true
    user_data = ";! /bin/bash\nnecho "Hello, World\" > index.html\nnnohup busybox httpd -f -p 8080&
    internet_max_bandwidth_in=10
    internet_max_bandwidth_out = 10
    internet_charge_type = "PayByTraffic"
    force_delete= true
}

resource "alicloud_ess_scaling_rule" "rule" {
    scaling_group_id = "${alicloud_ess_scaling_group.scaling.id}"
    adjustment_type = "TotalCapacity"
    adjustment_value = 2
    cooldown = 60
}
```
b) Run the `terraform plan` command to view resources.

c) After you confirm that the resources are correct, run the `terraform apply` command to create the Auto Scaling resources.
   After the resources are created, the public IP address of the SLB instance is displayed.
   After two minutes, Auto Scaling will create an ECS instance.

d) Enter and run the `curl http://<slb public ip>` command to verify whether you can access the Hello World page.
   If Hello, World is displayed, you can use the SLB instance to access the web page provided by the ECS instance.

5. Run the `terraform destroy` command to delete the test environment. With your confirmation, the entire deployment environment will be deleted.

You can use Terraform to easily delete environments and deploy new ones. To deploy a new environment, run the `terraform apply` command.