ALIBABA CLOUD

Alibaba Cloud

Elastic Compute Service

Elasticity

Document Version: 20220713

Alibaba Cloud

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

Table of Contents

1.Launch template

1.1. Create a launch template

This topic describes the notes for creating launch templates, how to create launch templates, and operations that can be performed by using launch templates.

Context

Before you create a launch template, take note of the following items:

- Up to 30 launch templates can be created per region within an account.
- 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.
- Launch templates cannot be modified after they are created. However, you can create versions for launch templates.

You can create launch templates in the Elastic Compute Service (ECS) console or on the ECS instance buy page. Alternatively, you can call the CreateLaunchTemplate operation by using Alibaba Cloud SDKs, OpenAPI Explorer, and Alibaba Cloud CLI to create launch templates.

Create a launch template in the ECS console

You can create launch templates beforehand to simplify the creation of ECS instances, scaling groups, and auto provisioning groups.

1.
2.
3.
4. On the Launch Templates page, click **Create Template**.
5. On the **Launch Template** page, configure the parameters in the Basic Configurations (Optional) and Advanced Configuration (Optional) steps.

   For more information about the parameters and their descriptions, see Create an instance by using the wizard.

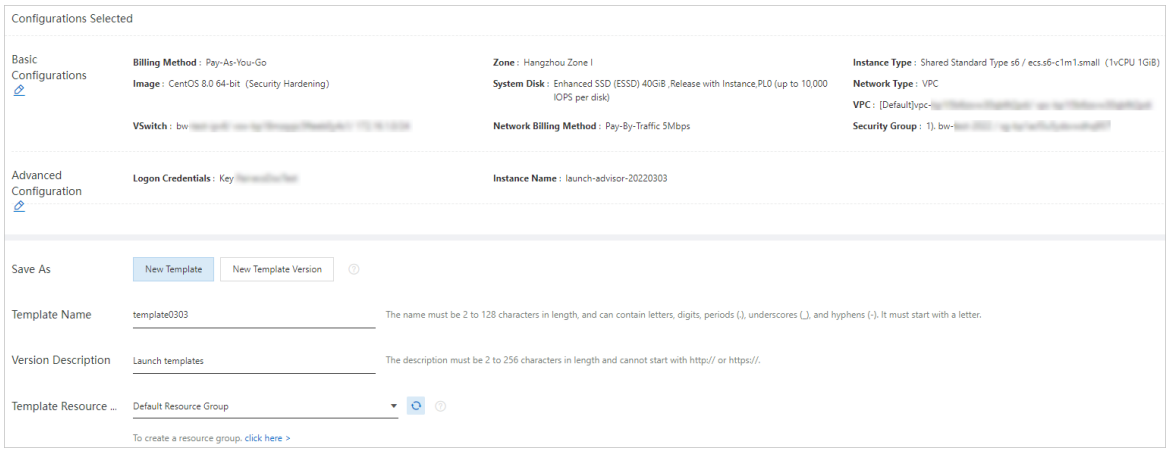
   > **Note** The first time you create a launch template, the **Clone Template** section is unavailable. If you have already created launch templates, you can select an existing launch template and one of its versions and then modify the configurations.

6. In the **Confirm Configuration** step, enter a template name and a template version description. Then, click **Create Launch Template**.
   - Selected configurations: You can click the icon in the **Basic Configurations** or **Advanced Configuration** section to modify the parameters.

     > **Note** The parameters in the **Basic Configurations** and **Advanced Configuration** sections are required to create instances and simplify subsequent instance creation. These parameters are optional and can be configured as needed.

- Template saving method: You can select the saving method of a new launch template based on your needs. If you select New Template in the Save As section, a launch template will be created, and the current configurations will be saved as the default version of the launch template. If you select New Template Version in the Save As section, a launch template version will be created, and the current configurations will be saved as the new version of the selected launch template.
- Launch template name and description: You can enter the name and description of the launch template for future management.
- Resource group of the launch template: You can select an existing resource group from the Template Resource Group drop-down list.

  If you want a new resource group, click **click here** to go to the Resource Group page to create a resource group. For more information, see Resource groups.



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

Create a launch template on the ECS instance buy page

If you want to save the configurations of an instance for simplifying subsequent instance creation, you can create a launch template when you create the instance.

1. Go to the Elastic Compute Service product page and click **Buy Now**.
2. On the **Custom Launch** tab, configure parameters.

   For more information about the parameters and their descriptions, see Create an instance by using the wizard.
3. In the **Preview** step, click **Save as Launch Template**.
4. In the Save as Launch Template dialog box, click **New Template**, enter a template name and a template version description, and then click **Save**.
5. In the **Created** message, click **View Template** to go to the ECS console and view the launch template that you have created.

Subsequent operations

After you create launch templates, you can use them to perform the following operations.

Operation	Description	References
Create an ECS instance	You can use an existing launch template to quickly create an instance. This eliminates the need to repeatedly configure parameters.	For more information, see Create an instance by using a launch template.
Create multiple ECS instances at a time	A launch template can work with the RunInstances operation to create multiple instances at a time. This eliminates the need to configure a large number of parameters to create multiple instances. You must specify the LaunchTemplateId and LaunchTemplateVersion parameters when you call the RunInstances operation.	For more information, see RunInstances.
Create a scaling group	You can use an existing launch template to quickly create a scaling group based on ECS instances. The system uses configurations defined in the launch template to create a scaling group. If some of the configurations do not fulfill business requirements, you can modify these configurations when you create the scaling group. For example, you can modify the virtual private cloud (VPC) and vSwitch required in the scaling configuration.	For more information, see Create a scaling group based on an existing ECS instance.
Create an auto provisioning group	Auto provisioning groups use specific versions of launch templates as instance configuration sources. Attributes such as instance images, security groups, and logon credentials from the launch templates are used by auto provisioning groups to create ECS instances. After an auto provisioning group is created, an ECS instance cluster is started and provisioned at the specified point in time, which improves the efficiency of offering a large number of ECS instances at a time.	For more information, see Create an auto provisioning group.

Related information

- CreateLaunchTemplate

1.2. Create a launch template version

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

Prerequisites

A launch template is created. For more information, see Create a launch template.

Context

Before you create a launch template version, take note of the following items:

- After a launch template is created, its launch template version is 1 by default. You can create more versions for the launch template.

> **Note** The number of launch template versions increments sequentially based on the order in which the versions are created. You cannot specify custom numbers for template versions.

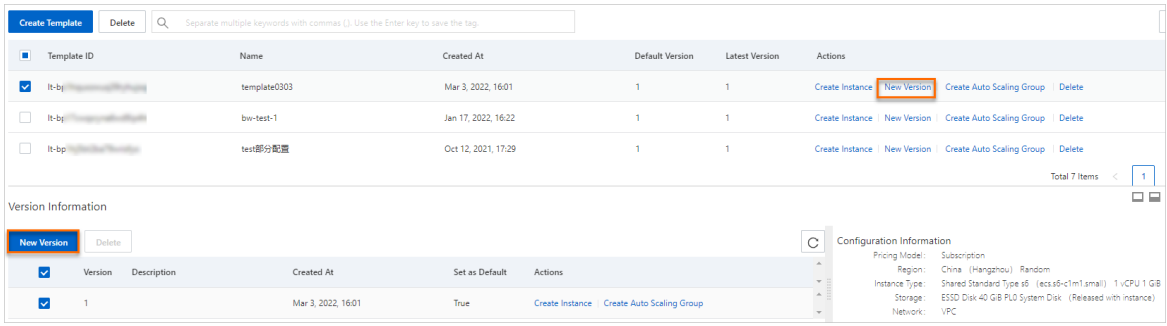
- A maximum of 30 versions can be created for each launch template.
- When you create a launch template version, all parameters are optional. However, if a launch template version does not contain required parameters such as the image or instance type, you must specify these parameters when you create an instance by using the launch template version.
- After a launch template version is created, its configurations cannot be modified.

Create a launch template version in the Elastic Compute Service (ECS) console

You can create a version for a launch template beforehand to simplify instance creation, regardless of whether you need to create instances for the moment.

1.
2.
3.
4. On the Launch Templates page, find the launch template for which you want to create a version and click **New Version** in the **Actions** column.

   Alternatively, click the ID of the launch template for which you want to create a version to view its configurations, and click **New Version** in the Version Information section.



5. On the **Launch Template** page, configure parameters.

   In the **Clone Template** section, select an existing template and version. Then, configure parameters based on the launch template version. For information about the parameters and their descriptions, see Create an instance by using the wizard.

6. In the **Confirm Configuration** step, select **New Template Version** in the Save As section and select a launch template for which you want to create a version from the Launch Template drop-down list.
7. Click **Create Launch Template**.
8. In the Created message, click **View New Version** to go to the ECS console and view the created version.

You can call the CreateLaunchTemplateVersion operation to create launch templates and launch template versions by using Alibaba Cloud SDKs, OpenAPI Explorer, and Alibaba Cloud CLI.

Create a launch template version on the Elastic Compute Service product page

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

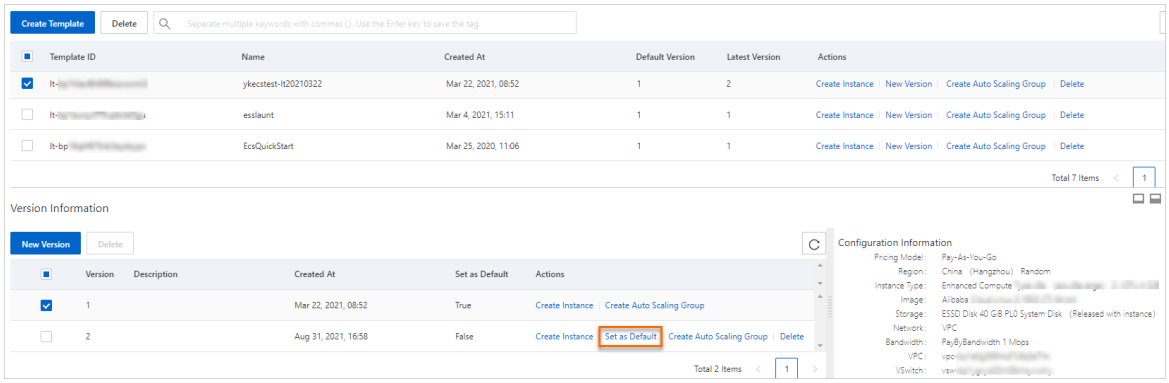
1. Go to the Elastic Compute Service product page and click **Buy Now**.
2. On the **Custom Launch** tab, configure parameters.

   For information about the parameters and their descriptions, see Create an instance by using the wizard.
3. In the **Preview** step, click **Save as Launch Template**.
4. In the Save as Launch Template dialog box, click **Create New Version** and select the launch template for which you want to create a version from the Launch Template drop-down list.
5. In the **Created** message, click **View New Version** to go to the ECS console and view the created version.

Change the default version of a launch template

If you use a particular launch template version frequently, you can specify it as the default version. This eliminates the need to specify a version when you use a launch template to create instances.

1.
2.
3.
4. Click the ID of the launch template whose default version you want to change. In the Version Information section, find the version that you want to set as the default version and click **Set as Default** in the **Actions** column.



You can call the ModifyLaunchTemplateDefaultVersion operation to change the default version of a launch template by using Alibaba Cloud SDKs, OpenAPI Explorer, and Alibaba Cloud CLI.

Related information

- CreateLaunchTemplateVersion
- ModifyLaunchTemplateDefaultVersion

1.3. Delete a launch template and a template version

This topic describes how to delete a launch template and a template version. If you delete a launch template, all versions of the launch template are also deleted. Proceed with caution when you delete a launch template.

Delete a template version

1.
2.
3.
4. Click the ID of the template whose version you want to delete.
5. In the **Version Information** section, select the template version that you want to delete and click **Delete** in the **Actions** column.

   > **Note** Default template versions cannot be deleted. If the template version that you want to delete is the default version, you can delete it only after you set another template version as the default version. If all versions of a launch template are no longer needed, you can delete the launch template.

6. In the Delete Version message, click **OK**.

Delete a launch template

1.
2.
3.
4. Find the launch template that you want to delete and click **Delete** in the **Actions** column.
5. In the Delete Template message, click **OK**.

Related information

- DeleteLaunchTemplateVersion
- DeleteLaunchTemplate

2.Create a scaling group based on an existing ECS instance

Auto Scaling scaling group

After you create a scaling group based on an Elastic Compute Service (ECS) instance, the system creates a default scaling configuration. This simplifies the preparations required to enable the scaling group and allows you to quickly put the scaling group in use.

Prerequisites

- The following requirements are met if you want to associate a Classic Load Balancer (CLB) instance with a scaling group:
- The following requirements are met if you want to associate an Application Load Balancer (ALB) server group with a scaling group:
  - The network type of the scaling group is virtual private cloud (VPC). The scaling group and the ALB server group that you want to associate are in the same VPC.
  - The ALB server group is in the Available state.
- The following requirements are met if you want to associate an ApsaraDB RDS instance with a scaling group:
  - One or more ApsaraDB RDS instances are created and are in the **Running** state. For more information, see What is ApsaraDB RDS?.
  - The scaling group and the ApsaraDB RDS instance are in the same region.

Context

ECS instances can be used as configuration sources to create scaling groups regardless of whether the instances use the subscription, pay-as-you-go, or preemptible instance billing method. After a scaling group is created based on an ECS instance, the system creates a default scaling configuration in which the billing method is pay-as-you-go.

> **Note** This default scaling configuration uses the image used by the ECS instance, instead of creating a custom image from the instance. If you want the scaling configuration to use an image that contains all the system configurations and data of an existing ECS instance, create a custom image from the ECS instance and then use the created custom image to update the scaling configuration. For more information, see Manually update images in scaling configurations.

After a scaling group is created, you can manually add existing ECS instances or configure ECS instances to be automatically added to the scaling group. Only pay-as-you-go and preemptible ECS instances can be automatically created for scaling groups, whereas subscription, pay-as-you-go, and preemptible instances can be manually added to scaling groups. For more information about limits on scaling groups, see Limits.

Procedure

1. Open the **Create Scaling Group** dialog box.
   i.
   ii.

iii.

iv.

v. Choose **More > Deployment & Elasticity > Create Scaling Group** in the **Actions** column corresponding to an ECS instance.

2. In the **Create Scaling Group** dialog box, configure parameters and click **OK**.

   If you create a scaling group from scratch, you must manually specify all parameters. If you create a scaling group based on an existing ECS instance, the following parameters are automatically specified:

   - **Instance Configuration Source**: is set to **Select Existing Instance**. The system extracts the configurations of your selected existing ECS instance to create a default scaling configuration and uses the scaling configuration as a template to create ECS instances in the scaling group. The extracted ECS instance configurations include the instance type, image, network type, security groups, logon password, and tags. If some of the ECS instance configurations such as the image do not meet your business requirements, you can modify the scaling configuration after the scaling group is created.
   - **Network Type**: is set to the network type of the selected ECS instance.
   - **VPC ID**: is set to the virtual private cloud (VPC) ID of the selected ECS instance if the instance is located in a VPC.
   - **VSwitch ID**: is set to the ID of the vSwitch to which the selected ECS instance is connected if the instance is located in a VPC.

   The following table describes the parameters required to create a scaling group. Accept the values automatically specified for parameters.

Parameter	Description
Scaling Group Name	The name of a scaling group must be 2 to 64 characters in length and can contain letters, digits, periods (.), underscores (_), and hyphens (-). The name must start with a letter or a digit.
Type	The type of instances that provide computing resources in the scaling group. Auto Scaling scales instances based on the Type parameter. Valid values: ○ ECS: ECS instances ○ ECI: elastic container instances

Parameter	Description
Instance Configuration Source	Auto Scaling creates instances based on the Instance Configuration Source parameter. Valid values: ○ **Launch Template**: A launch template contains information such as the key pair, RAM role, instance type, and network settings. A launch template does not contain passwords. Launch Template is available only if you set **Type** to **ECS**. If you use Launch Template, you must configure the Select Launch Template parameter and the Select Template Version parameter. To meet diverse business requirements, you can use the **Extend Launch Template Configurations** parameter to select multiple instance types. For more information about how to configure weights for instance types, see Use performance metrics to measure Auto Scaling. ○ **Select Existing Instance**: Select an existing instance. After you select an instance, Auto Scaling extracts the basic configurations of the instance to create a default scaling configuration. If you set **Type** to **ECS**, the basic configurations that are extracted from the selected ECS instance include the instance type, network type, security group, and base image. The instance logon password and tags are not extracted. The base image is the image used by the existing instance. The base image does not include instance data such as application data. If you want to include all system configurations of the instance and instance data in the scaling configuration, create a custom image for the instance and use the custom image to update the image of the scaling configuration. For more information, see Manually update images in scaling configurations. ○ **Create from Scratch**: Do not specify a template that is used to automatically create instances. After you create a scaling group, create a scaling configuration or specify a launch template. **Note** When you create a scaling group that contains ECS instances created in the ECS console, the instance configurations and network type of the instances are automatically populated. We recommend that you use the default settings.

Parameter	Description
Tag	You can add tags to help you find and manage scaling groups. For more information, see Overview. **Note** The tags that you add apply only to the scaling group. If you want to add tags to an instance in the scaling group, specify the tags in the scaling configuration or in the launch template.
Scale-In Policy	If you need to remove more than one instance from a scaling group, Auto Scaling removes instances based on the Scale-In Policy parameter. If multiple instances meet the conditions of the policy, a random instance is removed. The Scale-In Policy parameter is available only if you set **Type** to **ECS**. The Scale-In Policy parameter contains the **First Remove** and **Then Remove** fields. Specify different values for the two fields. The following part describes the values. **Note** If you set **Type** to **ECI**, Auto Scaling removes the instances that are created based on the earliest scaling configuration. Then, Auto Scaling removes the instances that are created at the earliest point in time from the results obtained based on the First Remove field. ◦ **Earliest Instance Created Using Scaling Configuration**: Auto Scaling removes instances that are created based on the earliest scaling configuration or launch template. No scaling configuration or launch template is associated with manually added instances. Therefore, manually added instances are not removed first. If more instances need to be removed from a scaling group after Auto Scaling removes all instances with which the earliest scaling configuration or launch template is associated, Auto Scaling removes manually added instances at random. **Note** Scaling Configuration in **Earliest Instance Created Using Scaling Configuration** specifies the instance configuration source that contains the scaling configuration and launch template. The version of a launch template does not indicate the sequence in which the template is added. For example, you use the lt-foress V2 template to create a scaling group, and then you replace the template with the lt-foress V1 template when you modify the scaling group. In this case, the scaling group considers the lt-foress V2 launch template as the earliest template. ◦ **Earliest Created Instance**: Auto Scaling removes the instances that are created at the earliest point in time. ◦ **Most Recent Created Instance**: Auto Scaling removes the instances that are created at the latest point in time. ◦ **No Policy**: This value is available only for the **Then Remove** field. If you select No Policy, Auto Scaling does not remove instances from the results obtained based on the First Remove field. If Auto Scaling removes instances based on the **Earliest Instance Created Using Scaling Configuration** value, you can select one of the following values for the Then Remove field:

Parameter	Description
	○ **No Policy**: Auto Scaling does not remove instances from the results obtained based on the First Remove field. ○ **Earliest Created Instance**: Auto Scaling removes the instances that are created at the earliest point in time from the results obtained based on the First Remove field. ○ **Most Recent Created Instance**: Auto Scaling removes the instances that are created at the latest point in time from the results obtained based on the First Remove field. **Note** The value of the **Scaling Policy** parameter affects how instances are removed from scaling groups. For more information about how to remove instances from scaling groups, see Configure a combination policy for removing instances.
Suspend Processes	You can suspend processes before you perform specific operations. For example, you can suspend the health check process before you stop an instance. This way, the instance is not removed from the scaling group if the health check fails. You can suspend the following processes for a scaling group: ○ **Scale-out**: If you suspend this process, Auto Scaling rejects all scale-out requests. ○ **Scale-in**: If you suspend this process, Auto Scaling rejects all scale-in requests. ○ **Health Check**: If you suspend this process, Auto Scaling suspends the health check process and does not remove unhealthy instances. ○ **Scheduled Task**: When the execution time of a scheduled task arrives, the scaling rules that are associated with the task are not triggered. ○ **Event-triggered Task**: When an event-triggered task enters the alert state, the scaling rules that are associated with the task are not triggered. For more information, see Suspend and resume scaling processes.
Deletion Protection	After you enable this feature, you cannot delete the scaling group by using the Auto Scaling console or by calling API operations.
Instance Health Check	After you enable this feature, Auto Scaling checks the status of instances on a regular basis. If an instance is not running, the instance is considered unhealthy and is removed from the scaling group. For more information, see ECS instance lifecycle in a scaling group.
Minimum Number of Instances	If the number of instances in a scaling group is less than the minimum number of instances allowed, Auto Scaling automatically creates instances until the number of instances in the scaling group reaches the minimum number.
Maximum Number of Instances	If the number of instances in a scaling group is greater than the maximum number of instances allowed, Auto Scaling automatically removes instances until the number of instances in the scaling group does not exceed the maximum number.

Parameter	Description
Expected Number of Instances	If you specify an expected number of instances, Auto Scaling automatically maintains the specified number of instances. For more information, see Expected number of instances. **Note** You can enable the Expected Number of Instances feature only when you create a scaling group. You cannot enable the Expected Number of Instances feature for a scaling group after you create it.
Default Cooldown Time (Seconds)	Specifies the default cooldown time of a scaling group. Unit: seconds. During the cooldown time, Auto Scaling rejects all requests for scaling activities triggered by event-triggered tasks. Scaling activities that are triggered by other types of tasks such as scheduled tasks and manually executed tasks are not subject to the cooldown time and can be immediately executed.
Network Type	The **Scaling Policy**, **Instance Reclaim Mode**, and **Associate ALB Server Group** parameters are available only if you set Network Type to VPC. **Note** When you create a scaling group that contains ECS instances created in the ECS console, the instance configurations and network type of the instances are automatically populated. We recommend that you use the default settings. A scaling group and instances in the scaling group must belong to the same network type. For example, if a scaling group resides in a VPC, the instances in the scaling group must also reside in the VPC. If a scaling group resides in the classic network, the instances in the scaling group must also reside in the classic network. **Note** After you create a scaling group, you cannot change the network type of the scaling group.
	The Scaling Policy parameter is available only if you set **Type** to **ECS** and **Network Type** to **VPC**. Valid values: ◦ **Priority Policy**: Instances are preferentially created in the zone where the vSwitch that has the highest priority resides. Auto Scaling preferentially scales instances in the zone where the vSwitch that has the highest priority resides. If the scaling fails, Auto Scaling attempts to scale instances in the zone where the vSwitch that has the next highest priority resides. **Note** If you set **Type** to **ECI**, Priority Policy is used. ◦ **Balanced Distribution Policy**: This policy is valid only if the scaling group is associated with multiple vSwitches that are distributed across more than two zones. Auto Scaling evenly distributes instances across the zones where the vSwitches reside based on this policy. If instances are not evenly distributed across multiple zones due to insufficient resources, you can use Balanced Distribution Policy to re-distribute instances across

Parameter	Description
	zones. For more information, see Rebalance the distribution of ECS instances.
Scaling Policy	○ **Cost Optimization Policy**: This policy is valid only if you specify multiple instance types in the scaling configuration. When a scale-out activity is triggered, Auto Scaling preferentially creates ECS instances that have the lowest vCPU price. When a scale-in activity is triggered, Auto Scaling preferentially removes ECS instances that have the highest vCPU price. If you select Preemptible Instance as the billing method in the scaling configuration, Auto Scaling preferentially creates preemptible instances. If preemptible instances cannot be created due to insufficient resources, Auto Scaling creates pay-as-you-go instances. If you select **Cost Optimization Policy**, configure the following parameters based on your business requirements: ○ **Minimum Pay-as-you-go Instances**: the minimum number of pay-as-you-go ECS instances in the scaling group. Default value: 0. If the number of pay-as-you-go ECS instances in the scaling group is less than the value of Minimum Pay-as-you-go Instances, Auto Scaling preferentially creates pay-as-you-go instances. ○ **Percentage of Pay-as-you-go Instances**: the percentage of pay-as-you-go ECS instances among all automatically created instances. Default value: 70%. When you calculate this percentage, the pay-as-you-go ECS instances do not include the minimum number of pay-as-you-go ECS instances that you specified for the scaling group. ○ **Lowest Cost Instance Types**: the number of the instance types that have the lowest price. Default value: 1. This parameter is valid only if multiple instance types are specified in the scaling configuration. Auto Scaling evenly creates preemptible ECS instances of the instance types that are provided at the lowest price. ○ **Enable Supplemental Preemptible Instances**: After you enable the Supplemental Preemptible Instances feature, Auto Scaling automatically creates preemptible instances five minutes before the existing instances are reclaimed. ○ **Use Pay-as-you-go Instances to Supplement Preemptible Capacity**: By default, this feature is enabled. After you enable this feature, Auto Scaling attempts to create pay-as-you-go instances to meet the required number of preemptible instances if preemptible instances cannot be created due to factors such as high prices and insufficient resources.
	The Instance Reclaim Mode parameter is available only if you set **Type** to **ECS** and **Network Type** to **VPC**. Valid values: ○ **Release**: Instances that are removed from the scaling group are released. Resources of these instances are not retained. During a scale-out activity, Auto Scaling creates new instances and adds the instances to the scaling group. **Note** If you set **Type** to **ECI**, instances that are removed from the scaling group are released by default. ○ **Economical Mode**: ECS instances that are removed from the scaling group are stopped and enter Economical Mode. Some resources of the ECS instances are retained, and you are charged for these resources.

Parameter	Description
Instance Reclaim Mode	During a scale-out activity, Auto Scaling preferentially adds the stopped ECS instances to the scaling group. After all stopped ECS instances are added, Auto Scaling determines whether to create ECS instances and add them to the scaling group based on your scale-out requirements. The Economical Mode setting can improve scaling efficiency. For more information, see Use the Economical Mode feature to scale instances faster. **Notice** ■ Your data stored on instances may be lost when the instances are reclaimed. To prevent data loss, do not store application data or logs on instances. ■ Stopped instances may be released due to the following reasons: ■ If the total number of instances in a scaling group exceeds the maximum number of instances allowed for the scaling group after you manually reduce the maximum number, Auto Scaling preferentially releases the ECS instances that are in the Stopped state. ■ If stopped instances fail to be added to a scaling group due to insufficient resources or overdue payments, the instances are released. ■ For more information about the Economical Mode setting, see the "Prerequisites", "Application resources", and "Trigger effects" sections in the Economical mode topic.

Parameter	Description
VPC	Select an existing VPC. **Note** When you create a scaling group that contains ECS instances created in the ECS console, the instance configurations and network type of the instances are automatically populated. We recommend that you use the default settings.
Select vSwitch	After you select a VPC, you must select a vSwitch. Each vSwitch resides in a single zone. To deploy instances across multiple zones, you must specify multiple vSwitches in different zones. We recommend that you select multiple zones to increase the success rate of scale-out. **Note** When you create a scaling group that contains ECS instances created in the ECS console, the instance configurations and network type of the instances are automatically populated. We recommend that you use the default settings.
Add Existing Instance	The Add Existing Instance parameter is available only if you set **Type** to **ECS** and set **Instance Configuration Source** to **Launch Template** or **Select Existing Instance**. If you specify an expected number of instances and then add existing instances to a scaling group, the expected number of instances for the scaling group automatically increases. For example, when you create a scaling group, you set Expected Number of Instances to 1 and add two existing instances to the scaling group. In this case, the expected number of instances is three. You can select **Enable the scaling group to manage the instance lifecycle**. ◦ If the scaling group manages the lifecycle of instances, the instances are automatically released when the instances are manually removed from the scaling group or are considered unhealthy. ◦ If the scaling group does not manage the lifecycle of instances, the instances are not automatically released when the instances are removed from the scaling group. **Note** You can add subscription instances to a scaling group. However, the lifecycle of the subscription instances cannot be managed by the scaling group.

Parameter	Description
Associate CLB Instance	After you associate a CLB instance with a scaling group, the instances that you add to the scaling group are automatically added as the backend servers of the CLB instance. Then, the CLB instance forwards requests to the instances. You can specify a server group to which you want to add instances. Valid values: ◦ Default server group: the group of instances that are used to receive requests. If you do not specify a vServer group or a primary/secondary server group for a listener, requests are forwarded to the instances in the default server group. ◦ vServer group: If you want to forward requests to different backend servers or configure domain name- or URL-based routing methods, you can use vServer groups. If you specify the default server group and multiple vServer groups at the same time, the instances are added to these server groups. **Note** You can associate only a limited number of CLB instances and vServer groups with a scaling group. To view the quota or request a quota increase, go to the Quota Center.
Associate ALB Server Group	The Associate ALB Server Group parameter is available only if you set **Network Type** to **VPC**. After you associate an ALB server group with a scaling group, the instances that you add to the scaling group are automatically added to the ALB server group to process requests that are forwarded by the ALB instance. You must specify the port number and weight for each backend server. By default, the weight of a backend server is 50. If you increase the weight of a server, the number of requests that are forwarded to the server increases. If you set the weight to 0, no requests are forwarded to the server. If you associate multiple ALB server groups with the same scaling group, all instances that you add to the scaling group are added to the server groups. **Note** You can associate only a limited number of ALB server groups with a scaling group. To view the quota or request a quota increase, go to the Quota Center.
Associate ApsaraDB RDS Instance	The Associate ApsaraDB RDS Instance parameter is available only if you set **Type** to **ECS**. After you associate an ApsaraDB RDS instance with a scaling group, the internal IP addresses of ECS instances that you add to the scaling group are automatically added to the whitelist that manages access to the ApsaraDB RDS instance to allow internal communication. **Note** You can associate only a limited number of ApsaraDB RDS instances with a scaling group. To view the quota or request a quota increase, go to the Quota Center.

Parameter	Description
Configure Notification	When a scaling activity succeeds, fails, or is rejected, Auto Scaling sends notifications to you by using text messages, internal messages, or emails. For more information, see Set notification receiving.

3. In the **Create Scaling Group** dialog box, click **OK**.
4. On the **Scaling Groups** page, find the scaling group that you created and choose > **Enable** in the **Actions** column.
   After the scaling group is enabled, Auto Scaling maintains a collection of ECS instances based on the settings of the scaling group such as the instance configuration source and minimum number of instances. For example, if the minimum number of instances is set to a non-zero value, Auto Scaling creates a corresponding number of ECS instances in the scaling group to ensure that the actual number of instances in the scaling group is greater than or equal to the minimum number.

What's next

You can manually add, remove, or delete ECS instances for the scaling group. Alternatively, you can configure scheduled tasks or event-triggered tasks to automatically scale the scaling group. For more information, see the following topics:

- Manually add an ECS instance to a scaling group
- Manually remove or delete an ECS instance
- Automatically add ECS instances
- Automatically remove ECS instances

Related information

- Create a scaling group
- CreateScalingGroup

3.Resource Orchestration Service

3.1. ROS overview

Resource Orchestration Service (ROS) is a service provided by Alibaba Cloud to simplify the management of cloud computing resources. The ROS engine automatically creates and configures all resources in a stack based on a template, making automatic delivery of ECS and ApsaraDB for RDS instances possible.

For more information, see What is ROS?

Features

- Repeated deployment

  You can use the same template to deploy resources in the development, test, or production environment by specifying different values for parameters. For example, you can set the number of ECS instances in the test environment to 2 and the number of ECS instances in the production environment to 20. You can also use the same template to deploy resources in multiple regions. This improves the efficiency of multi-region deployment.

- Standardized deployment

  In practice, subtle differences in different environments often lead to complicated management, high costs, and prolonged troubleshooting time. By using ROS for repeated deployment, you can standardize deployment environments, minimize the differences between environments, and integrate environment configurations into templates.

- Fully managed automation

  You do not need to purchase or maintain the resources that are used to execute templates. You only need to focus on the resources required by your business and the template specifications. If you want to create multiple projects that are distributed across multiple stacks, the fully managed automation service allows you to create tasks faster.

- Authentication and audit

  ROS is integrated with Resource Access Management (RAM) to provide unified authentication. This eliminates the need to establish user authentication and permission systems. You can use ActionTrail to review all O&M operations of Alibaba Cloud services, including operations on ROS.

Benefits

- Infrastructure as Code

  ROS is an Infrastructure as Code (IaC) solution provided by Alibaba Cloud to quickly implement IaC as a key component of DevOps.

- Efficiency improvement

  ROS provides solution templates to reduce nearly 90% of deployment time for complex solutions such as SAP deployment. You can also use templates to implement repeated deployment in a standardized manner to improve efficiency.

- Architecture optimization

ROS supports one-click deployment of classic cloud migration solutions, which simplifies the cloud migration process and optimizes cloud architecture.

- Internal compliance control

  ROS templates can be used to deploy a predefined cloud environment, which simplifies financial and IT compliance audits.

- Cost-effectiveness

  The preconfigured ROS templates can be used to deploy or release applications and cloud environments on a regular basis to implement on-demand usage and pay-as-you-go billing.

Usage

You can create a stack template in the ROS console or by calling API operations. Then, you can use the template to quickly create and manage resources. For more information, see the following topics in the *ROS documentation*:

- Template structure
- Create a stack
- List of API operations by function

You can also perform the following operations:

- Uses Git or Subversion (SVN) to manage template versions and then calls ROS API operations to maintain stacks.
- Uses Alibaba Cloud command-line interface (CLI) to create stacks. For more information, see Stack operations.

4.Deployment sets

4.1. Overview

A deployment set is a policy that controls the distribution of Elastic Compute Service (ECS) instances. You can use deployment sets to design how to implement disaster recovery and service availability when you create ECS instances.

Deployment policy

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

Deployment sets support the following policies:

- Deployment set-based high availability policy

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

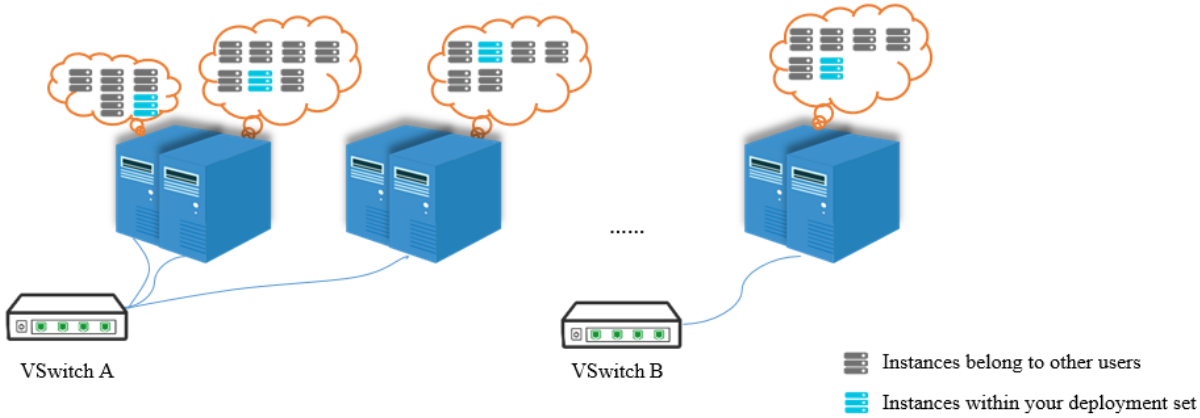
- Deployment set group-based high availability policy

  When you use the deployment set group-based high availability policy, each deployment set can be divided into up to seven deployment set groups. Multiple instances in a deployment set can be distributed to different deployment set groups as needed. Instances in different deployment set groups are strictly distributed to different physical servers. Instances in the same deployment set group are not guaranteed to be strictly distributed to different physical servers.

> **Note** If a supply shortage occurs in a specified region, you may not be able to create ECS instances or restart stopped pay-as-you-go instances that are in economical mode in that region. In these cases, we recommend that you wait a while and try again.

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 keep the network types of the instances the same. For example, you can select the same virtual private cloud (VPC) for the ECS instances when you create them.

Billing

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

Limits

Before you use deployment sets, take note of the following items:

- Deployment sets cannot be merged.
- You cannot create preemptible 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 20 ECS instances within each zone. This limit varies with your ECS usage. 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: *20 × Number of zones within the region*.
- Instances of the following instance families can be created in deployment sets:
  - c7, g7, r7, c6, g6, r6, c5, g5, r5, c6e, g6e, r6e, c7se, g7se, r7se, r6se, c7t, g7t, r7t, c7a, g7a, r7a, c6a, g6a, r6a, g5ne, re6, re4, and ic5
  - hfc7, hfg7, hfr7, hfc6, hfg6, hfr6, hfc5, and hfg5
  - d2s, d2c, d1, d1ne, d1-c14d3, and d1-c8d3
  - i3, i3g, i2, i2g, i2ne, i2gne, and i1
  - se1ne, sn1ne, sn2ne, and se1
  - ebmg5, sccgn6, scch5, sccg5, scch5s, and sccg5s
  - s6, t6, xn4, mn4, and n4
  - gn6i

  You can call the DescribeDeploymentSetSupportedInstanceTypeFamily operation to query instance families that support deployment sets.

- Supply shortage may result in a failure to create an instance or restart a pay-as-you-go instance that is in economical mode in a deployment set. For more information, see Economical mode.

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

References

- Create a deployment set
- Create an ECS instance in a deployment set
- Change the deployment set of an instance
- Manage deployment sets
- Delete a deployment set

API operations

- Create a deployment set: CreateDeploymentSet.
- Add an instance to a deployment set or migrate an instance from one deployment set to another: ModifyInstanceDeployment.
- Query deployment sets: DescribeDeploymentSets.
- Modify the attributes of a deployment set: ModifyDeploymentSetAttribute.
- Delete a deployment set: DeleteDeploymentSet.

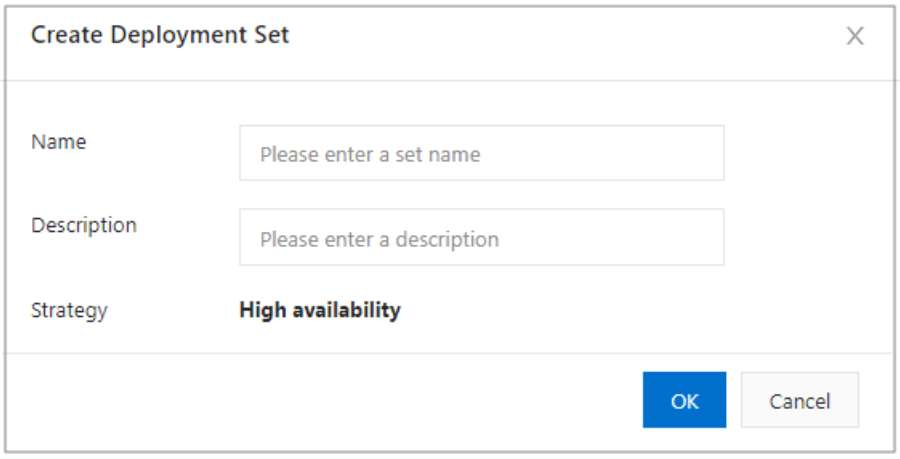
4.2. Create a deployment set

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 tolerance.

Procedure

1.
2.
3.
4. On the **Deployment Sets** page, click **Create Deployment Set**.
5. In the **Create Deployment Set** dialog box, set **Name** and **Description**.

   The **Strategy** parameter supports only the **High Availability** option. For more information about the deployment set strategy, see Overview.



6. Click **OK**.

What's next

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

- Create an instance in the deployment set. For more information, see Create an ECS instance in a deployment set.

Related information

- CreateDeploymentSet

4.3. Create an ECS instance in a deployment set

This topic describes how to create an Elastic Compute Service (ECS) instance in a deployment set by using the ECS console.

Prerequisites

A deployment set is created in a specified region. For more information, see Create a deployment set.

Context

When you create ECS instances in a deployment set, you can create up to 20 ECS instances within 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: 20 × Number of zones within the region. These limits vary with your ECS usage.

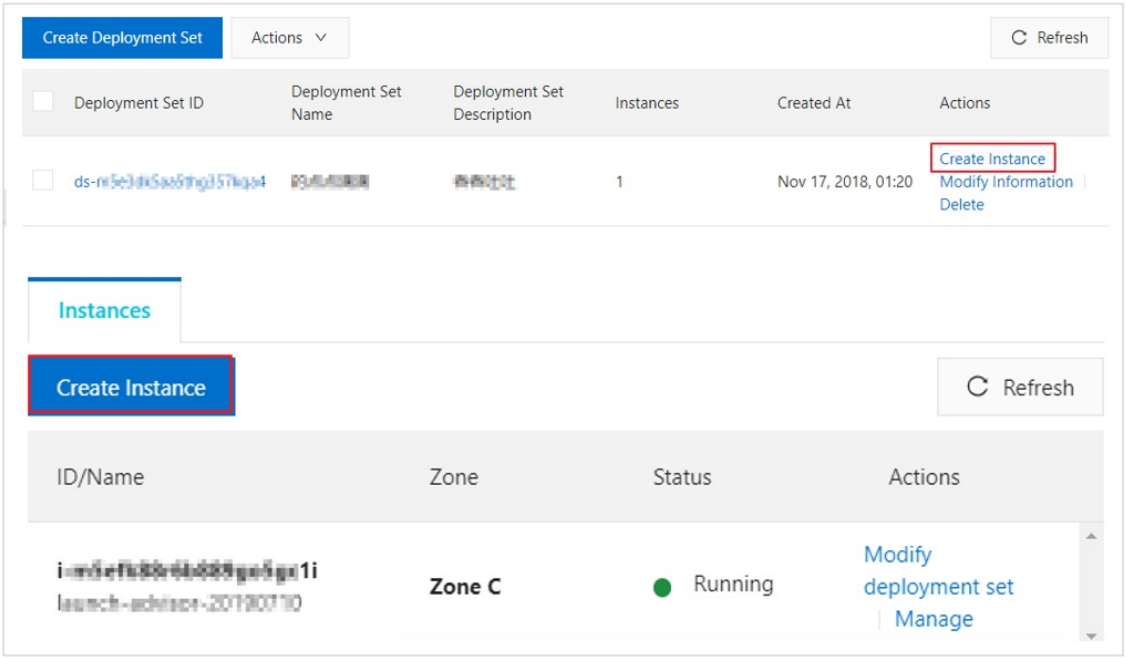
When you create an instance, you can use a launch template or use the batch creation feature to facilitate instance creation. For more information, see 实例启动模板概述.

Procedure

1.
2.
3.
4. On the Deployment Sets page, find the deployment set in which you want to create an instance.

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 the deployment set ID. In the **Instances** panel, click **Create Instance**.

5. On the page that appears, complete the instance configurations on the **Custom Launch** tab.

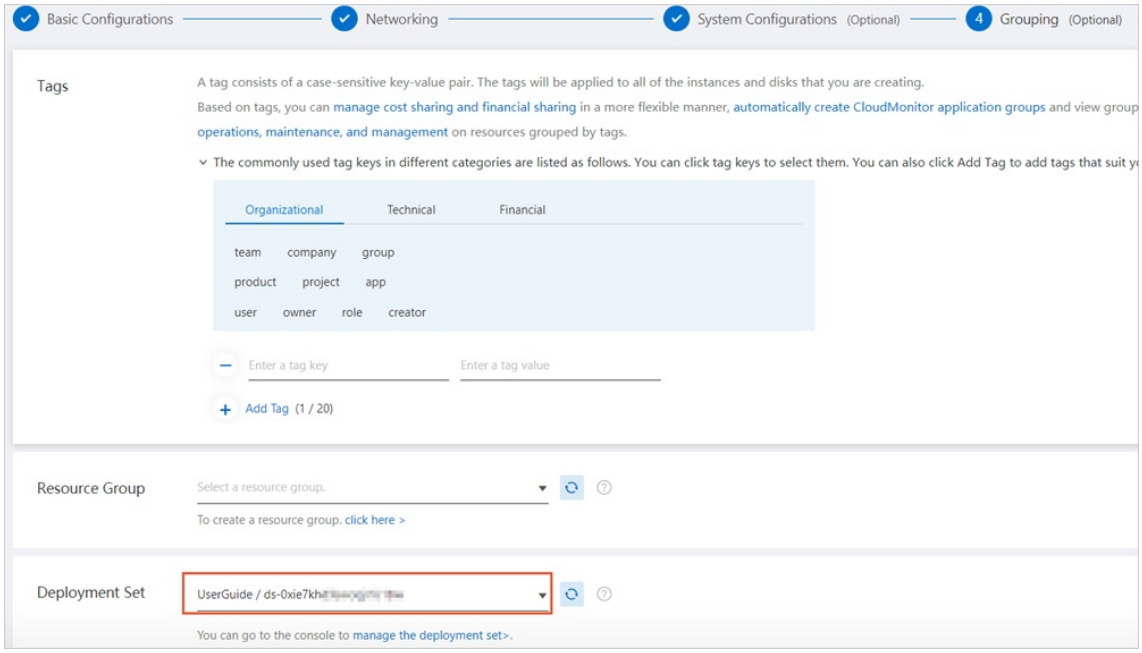
   For more information, see Create an instance by using the wizard. Take note of the following configurations when you create an instance:

   - **Basic configurations**:
     - **Region**: Select a region in which the deployment set is located. The ECS instance and the deployment set must be located in the same region.
     - **Zone**: Select a zone. Up to 20 ECS instances can be created in each zone of a deployment set.
     - **Instance Type**: Select an instance family that supports deployment sets. The following instance families of ECS instances support deployment sets:
       - c7, g7, r7, c6, g6, r6, c5, g5, r5, c6e, g6e, r6e, c7se, g7se, r7se, r6se, c7t, g7t, r7t, c7a, g7a, r7a, c6a, g6a, r6a, g5ne, re6, re4, and ic5
       - hfc7, hfg7, hfr7, hfc6, hfg6, hfr6, hfc5, and hfg5
       - d2s, d2c, d1, d1ne, d1-c14d3, and d1-c8d3
       - i3, i3g, i2, i2g, i2ne, i2gne, and i1
       - se1ne, sn1ne, sn2ne, and se1
       - ebmg5, sccgn6, scch5, sccg5, scch5s, and sccg5s
       - s6, t6, xn4, mn4, and n4
       - gn6i

     You can call the DescribeDeploymentSetSupportedInstanceTypeFamily operation to query instance families that support deployment sets.

- (Optional) **Quantity**: 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.

- (Optional) **System Configurations (Optional) > Sequential Suffix**: Add sequential suffixes to instance names and hostnames after you create multiple instances. A sequential suffix ranges from 001 to 999.
- **Grouping > Deployment Set**: Select the deployment set in which to create the ECS instance.



- (Optional) **Preview > Save as Launch Template**: Save your configurations as a launch template that you can use to create an instance the next time. For more information, see 实例启动模板概述.

6. Check the settings you have made and 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 information

- RunInstances

4.4. Change the deployment set of an instance

This topic describes how to add an Elastic Compute Service (ECS) instance to a deployment set or migrate an instance from one deployment set to another.

Prerequisites

The instance is in the **Stopped** or **Running** state.

Procedure

1.
2.
3.
4. Find the instance for which you want to change the deployment set and choose **More > Instance Settings > Change Deployment Set** in the **Actions** column.
5. In the **Change Deployment Set** dialog box, select the destination deployment set and set the Force Change parameter.

   Valid values of Forced Change:

   - **Yes**: allows the instance to be migrated to another host, which may cause the instance to restart.
   - **No**: does not allow the instance to be migrated to another host. The instance must remain on the current host, which may cause a failure to change the deployment set of the instance.
6. Click **OK**.

Related information

- ModifyInstanceDeployment

4.5. 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.

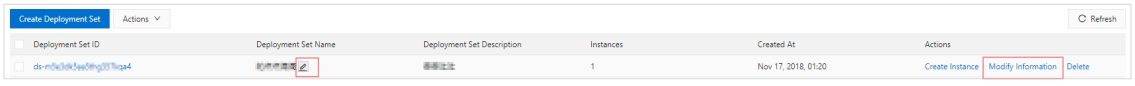
Edit deployment set information

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

1.
2.
3.
4. Find the deployment set that needs to be modified.
5. 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.



6. 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.
2.
3.
4. Select one or more deployment sets that need to be deleted, hover the cursor over the **Actions** menu, and then click **Delete**.

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

You can use the DeleteDeploymentSet API operation to delete deployment sets.

4.6. Delete a deployment set

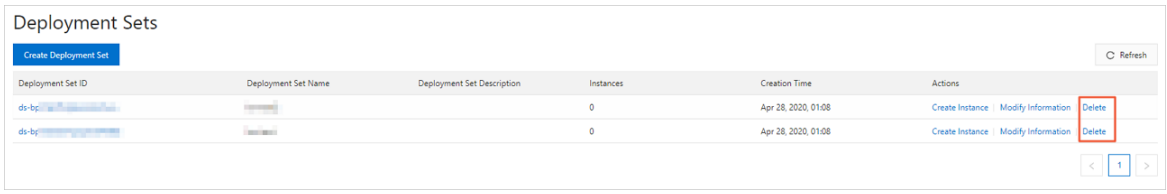
You can delete any deployment sets that you no longer need to ensure that the usage limit is not exceeded.

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.
2.
3.
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 information

- DeleteDeploymentSet

5.Manage auto provisioning groups

5.1. Overview

Auto Provisioning is a service that enables fast provisioning of Elastic Compute Service (ECS) instances. You need only to make simple configurations to automate the creation of instances that use different billing methods (pay-as-you-go and preemptible instances) across instance types and zones, which improves the efficiency of batch creating a large number of instances. Auto Provisioning provides a variety of provisioning policies to alleviate the impacts on your business caused by the reclaiming of preemptible instances, so that you can use computing power at low costs.

Introduction

Auto provisioning groups automatically create instances based on your specified resource pools, target capacities, and provisioning policies. Auto provisioning groups eliminate the need to track the creation processes or calculate the costs of individual instances. Auto provisioning groups have the following attributes:

- Resource pool: A resource pool consists of a single zone and a single instance type. You can specify multiple zones and instance types to make multiple resource pools available for use to create instances.
- Target capacity: Target capacity is the computing power that is scheduled to be provisioned, and can include the capacity of preemptible instances and pay-as-you-go instances. You can specify the target capacity based on instances, vCPUs, or memory size.

  By default, auto provisioning groups use preemptible instances to meet the target capacity. You can also specify to use pay-as-you-go instances to provide the minimum required computing power. If you specify to use the capacity of pay-as-you-go instances, pay-as-you-go instances are created with priority over preemptible instances. Then, preemptible instances are created to make up for the remaining target capacity. If preemptible instances cannot be created due to insufficient resources, pay-as-you-go instances are created to meet the target capacity.
- Provisioning policy: A provisioning policy is used to choose resource pools to use to create instances. For example, you can choose the resource pool with the lowest unit price to reduce your costs, or choose to create instances in different zones to improve the availability of your instances.

Scenarios

Similar to preemptible instances, auto provisioning groups are applicable to stateless applications such as scalable website services, image rendering, big data analytics, and parallel computing. For more information, see Overview.

Usage notes

Before you create an auto provisioning group, make sure that the following operations are performed:

- Specify the launch template that contains the basic instance configurations. For more information, see 实例启动模板概述.

The auto provisioning group uses attributes such as the image, security group, and logon credential specified in the launch template, but not the instance type and the vSwitch that specifies the zone where to create the instances. The auto provisioning group uses a separately specified vSwitch and instance type instead.

- Specify the resource pools to use to create instances across zones and instance types.

  A resource pool consists of a single zone and a single instance type. If resources in one resource pool are insufficient, the auto provisioning group tries another resource pool with sufficient resources to create instances. A larger number of resource pools increases the available options for creating instances and improves the success rate of creating instances.

  > **Note** You can specify only a single vSwitch within a single zone. If you specify multiple vSwitches within the same zone, only the first vSwitch takes effect.

- Specify the target capacity of the auto provisioning group and the proportions of capacities of different types.

  The target capacity can be specified based on instances, vCPUs, or memory size. Auto Provisioning uses the weight of an instance type to indicate the capacity of a single instance of this instance type. The following rules apply to weights:

  - If the target capacity is specified based on instances, the weights of all instance types are the same.
  - If the target capacity is specified based on vCPUs, the weight of an instance type depends on the number of vCPUs. The more vCPUs that an instance type has, the higher the weight of the instance type and the fewer instances of the instance type that are required to meet the target capacity.
  - If the target capacity involves multiple instance type factors such as vCPUs and memory, you must evaluate the computing power that each specified instance type is able to contribute to the target capacity and set a weight for each instance type. A larger weight indicates that the instance type is able to contribute more computing power.

    > **Note** You can specify the weights of instance types only when you create auto provisioning groups by calling the CreateAutoProvisioningGroup operation.

  If you use pay-as-you-go instances to meet the minimum computing power requirements and use preemptible instances to meet the remaining target capacity, your costs can be significantly reduced.

- Specify the provisioning policy.

  The following table describes the provisioning policies in Auto Provisioning.

Policy	Applicable to	Method	Description
Capacity optimization policy (capacity-optimized)	Preemptible instances	◦ ECS console (preemptible instance): Set Provisioning Policy to **Capacity Optimization Policy**. ◦ API operation (preemptible instance): Set SpotAllocationStrategy to *capacity-optimized*.	The resource pool that is most cost-effective and that has the highest success rate of instance creation is used based on the prices and reclaim rates to create preemptible instances. This can effectively reduce the possibilities of preemptible instances being reclaimed and ensure stable capacity.

Policy	Applicable to	Method	Description
Balanced distribution policy (diversified)	Preemptible instances	◦ ECS console (preemptible instance): Set Provisioning Policy to **Balanced Distribution Policy**. ◦ API operation (preemptible instance): Set SpotAllocationStrategy to *diversified*.	Resource pools in multiple zones are evenly used to create instances. This eliminates the possibilities of instance creation failures caused by insufficient resources within a single zone and can effectively improve the disaster recovery capabilities of applications.
Cost optimization policy (lowest-price)	Preemptible and pay-as-you-go instances	◦ ECS console (preemptible instance): Set Provisioning Policy to **Cost Optimization Policy**. ◦ API operation (preemptible instance): Set SpotAllocationStrategy to *lowest-price*. ◦ API operation (pay-as-you-go instance): Set PayAsYouGoAllocationStrategy to *lowest-price*.	The resource pool with the lowest cost is used to create instances. This can effectively reduce costs. The resource pool with the lowest cost has vCPUs whose unit prices are the lowest. If you create an auto provisioning group by calling the CreateAutoProvisioningGroup operation, you can specify to use the cost optimization policy for preemptible instances and specify the number of resource pools to use by setting the SpotInstancePoolsToUseCount parameter. You can use multiple resource pools with the lowest costs to create instances. For example, you specify 100 preemptible instances as the target capacity of an auto provisioning group and set SpotInstancePoolsToUseCount to *5*. Each resource pool is used to create 20 instances. When preemptible instances created by using one resource pool are reclaimed, preemptible instances created by using other resource pools are still available. This can effectively improve the availability of services.

Policy	Applicable to	Method	Description
Priority-based policy (prioritized)	Pay-as-you-go instances	API operation (pay-as-you-go instance): Set PayAsYouGoAllocationStrategy to *prioritized*.	Resource pools are used based on their priorities in descending order. When resources in a high-priority resource pool are insufficient, lower-priority resource pools are used. You must call the CreateAutoProvisioningGroup operation and set the **LaunchTemplateConfig.N.Priority** parameter to specify the priority-based policy and the priority of the resource pool.

> **Note** You can specify provisioning policies for pay-as-you-go instances only when you call the CreateAutoProvisioningGroup operation to create auto provisioning groups. In other cases, the cost optimization policy is used by default.

- Specify the maximum hourly prices.

  You can specify the maximum hourly prices for preemptible instances for a single resource pool or for all resource pools. If the specified maximum prices are exceeded, the auto provisioning group stops creating preemptible instances even if the target capacity is not met. This ensures that your provisioned resource costs remain within your budget.

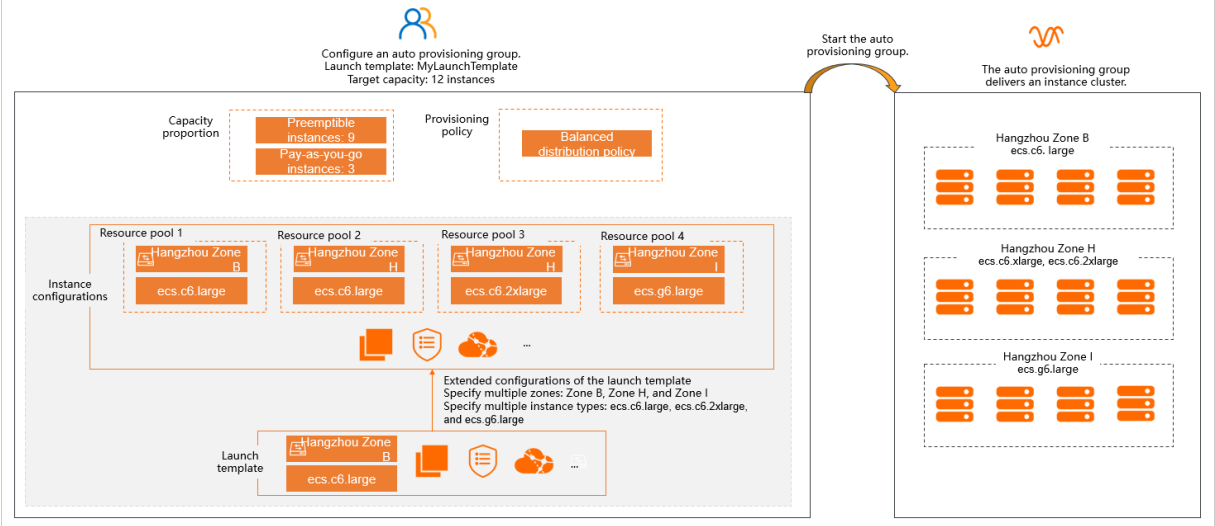
  You can use discount plans such as reserved instances and savings plans to reduce your costs of pay-as-you-go instances. For more information, see Overview and Overview.

- Specify whether to maintain the target capacity.

  The type of an auto provisioning group can be set to One-time Delivery or Continuous Delivery and Maintain Capacity. If you select Continuous Delivery and Maintain Capacity when you create an auto provisioning group, the auto provisioning group automatically checks the health status of instances and compares the real-time and target capacities. When instances are removed because they are unhealthy or due to insufficient resources, the auto provisioning group creates instances to maintain the target capacity to meet your computing power needs.

After an auto provisioning group starts, it chooses appropriate resource pools based on the specified provisioning policy to provision instances that meet the target capacity. For example, if you want to provision a cluster that contains 12 instances by using the MyLaunchTemplate launch template and want improved availability of the cluster, you can perform the following procedure to configure the auto provisioning group:

1. Specify multiple resource pools based on the launch template.
2. Specify the proportions of capacities of preemptible and pay-as-you-go instances.
3. Specify the provisioning policy.



Billing

Auto Provisioning is available free of charge. However, you are charged for instances that are created. Auto Provisioning supports preemptible and pay-as-you-go instances. For more information about billing, see Overview and Pay-as-you-go.

> **Warning** Make sure that you have sufficient balance within your account. If you have overdue payments within your account, all pay-as-you-go and preemptible instances are stopped. For more information, see Settlement cycle. In this case, the auto provisioning group cannot create instances. The auto provisioning group determines the stopped instances to be unhealthy based on the health check results and then removes and releases these instances.

Limits

- Auto provisioning groups cannot provision instances across regions.
- For each auto provisioning group, you can specify the specified version of a single launch template as the basic configurations of instances. However, you can specify multiple resource pools based on the instance type specified in the launch template.
- Each auto provisioning group can include a maximum of 20 resource pools. Each resource pool is a combination of a single zone and a single instance type.
- A maximum of 1,000 instances can be created in each auto provisioning group.

5.2. Manage the service linked role for Auto Provisioning

This topic describes how to use the service linked role for Auto Provisioning to grant Auto Provisioning the permissions on Alibaba Cloud resources.

Prerequisites

If you are a RAM user, you are granted permissions to use Auto Provisioning so that you can manage the service linked roles for Auto Provisioning. For more information, see Grant permissions to a RAM user.

The trusted policy of the service linked role includes the following content:

> **Note** Replace <Account ID> with the ID of your Alibaba Cloud account.

```
{
    "Statement": [
        {
            "Action": [
                "ram:CreateServiceLinkedRole"
            ],
            "Resource": "acs:ram:*:<account ID>:role/*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ram:ServiceName": [
                        "autoprovisioning.ecs.aliyuncs.com"
                    ]
                }
            }
        }
    ],
    "Version": "1"
}
```

Context

AliyunServiceRoleForAutoProvisioning is a service linked role provided by Resource Access Management (RAM) to Auto Provisioning. Auto Provisioning can use AliyunServiceRoleForAutoProvisioning to obtain the access to Elastic Compute Service (ECS), Virtual Private Cloud (VPC), ApsaraDB RDS, Server Load Balancer (SLB), Operation Orchestration Service (OOS), Message Service (MNS), and Cloud Monitoring Service (CMS). For more information about service linked roles, see Service-linked roles.

Create AliyunServiceRoleForAutoProvisioning

When you create an auto provisioning group, the system checks whether your Alibaba Cloud account has the AliyunServiceRoleForAutoProvisioning service linked role. If your account does not have the role, the system prompts you to create the role. After you confirm, the system automatically creates the role. You can also manually create AliyunServiceRoleForAutoProvisioning. For more information, see Create a service-linked role.

The permissions of service linked roles are defined and used by the corresponding cloud services. You cannot add, modify, or delete permissions for service linked roles. You can view role permissions on the role details page. For more information, see View the basic information about a RAM role.

Delete AliyunServiceRoleForAutoProvisioning

If you no longer want to use AliyunServiceRoleForAutoProvisioning, such as when you do not want to use an auto provisioning group to create and manage resources or if you want to know how deletion of the role will impact your business, you can delete the role. For more information, see Delete a RAM role.

> **Note** Before you can delete AliyunServiceRoleForAutoProvisioning, you must delete the auto provisioning groups in all regions in the current account. Otherwise, the deletion will fail.

After AliyunServiceRoleForAutoProvisioning is deleted, you cannot use Auto Provisioning to create or manage resources.

5.3. Create an auto provisioning group

This topic describes how to create an auto provisioning group in the Elastic Compute Service (ECS) console. The auto provisioning group can create instance clusters based on your configurations.

Prerequisites

- Your account is granted permissions on Auto Provisioning.

  > **Note** The first time you access the Auto Provisioning page, you must follow the instructions to assume the `AliyunECSAutoProvisioningGroupRole` Resource Access Management (RAM) role.

- A launch template is created. For more information, see Create a launch template.

  Auto provisioning groups use specific versions of launch templates as instance configuration sources. Properties such as instance images, security groups, and logon credentials from the launch templates are used by auto provisioning groups to create instances.

Procedure

1. Go to the Auto Provisioning page.
   i.
   ii.
   iii.
   iv.
2. Click **Create Auto Provisioning Group**.
3. In the **Group Name** field, enter the name for 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, underscores (_), hyphens (-), and periods (.).
4. In the **Target Capacity** section, configure parameters related to the capacity of the auto provisioning group.

   The capacity-related parameters determine the sum of computing power provisioned by the auto provisioning group and the proportions of computing power provided by preemptible and pay-as-you-go instances. The following table describes these parameters.

Parameter	Description

Parameter	Description
Target Capacity	The computing power that the auto provisioning group is scheduled to provision. You can specify this capacity based on **Instances** or **vCPUs**. After you select **Use Pay-as-you-go Instances to Provide Computing Power**, you can specify the computing power provided by the pay-as-you-go instances. By default, only preemptible instances are created. Pay-as-you-go instances are created only after you set Pay-as-you-go Instance Capacity to a value greater than 0. **Note** If the target capacity involves multiple instance type factors such as vCPUs and memory, call the CreateAutoProvisioningGroup operation to create an auto provisioning group and set weights for the specified instance types.
Pay-as-you-go Instance Capacity	The target capacity of pay-as-you-go instances that the auto provisioning group is scheduled to provision. You can specify this number of instances. You can use pay-as-you-go instances to ensure that the lowest computing power requirement can be met because preemptible instances may be reclaimed.

Auto Provisioning uses the weight of an instance type to indicate the capacity of a single instance of this instance type in an auto provisioning group.

- If the target capacity is specified based on **Instances**, the weights of all instance types are the same.
- If the target capacity is specified based on **vCPUs**, the weight of an instance type depends on the number of vCPUs. The more vCPUs that an instance type has, the higher the weight of the instance type and the fewer instances of the instance type that are required to meet the target capacity. The following table describes an example in which different weights are set for three instance types based on the number of vCPUs.

Instance type	vCPU	Weight
ecs.c6.large	2	2
ecs.c6.xlarge	4	4
ecs.c6.2xlarge	8	8

Note When you create an auto provisioning group in the ECS console, you do not need to set a weight for each instance type. Weights are automatically assigned based on the number of vCPUs of each instance type.

If the target capacity is 24 vCPUs, the auto provisioning group can deliver instances of one or more of the ecs.c6.large, ecs.c6.xlarge, and ecs.c6.2xlarge instance types based on their weights to meet the target capacity. Examples:

- 12 ecs.c6.large instances
- 8 ecs.c6.large instances and 1 ecs.c6.2xlarge instance

- 4 ecs.c6.large instances, 2 ecs.c6.xlarge instances, and 1 ecs.c6.2xlarge instance

- If the target capacity involves multiple instance type factors such as vCPUs and memory, evaluate the computing power that each specified instance type is able to contribute to the target capacity and set a weight for each instance type. A larger weight indicates that the instance type is able to contribute more computing power. For example, an application requires an instance cluster that provides a total computing power of 20 vCPUs and 48 GiB memory and also requires that the computing power of each instance is a multiple of 2 vCPUs and 4 GiB memory. You can set the target capacity of the auto provisioning group to 48 and set a weight for each instance type, as shown in the following table.

Instance type	vCPU	Memory	Weight
ecs.c6.large	2	4 GiB	4
ecs.c6.xlarge	4	8 GiB	8
ecs.c6.2xlarge	8	16 GiB	16

  The auto provisioning group can deliver instances of one or more of the ecs.c6.large, ecs.c6.xlarge, and ecs.c6.2xlarge instance types based on their weights to meet the target capacity. Examples:

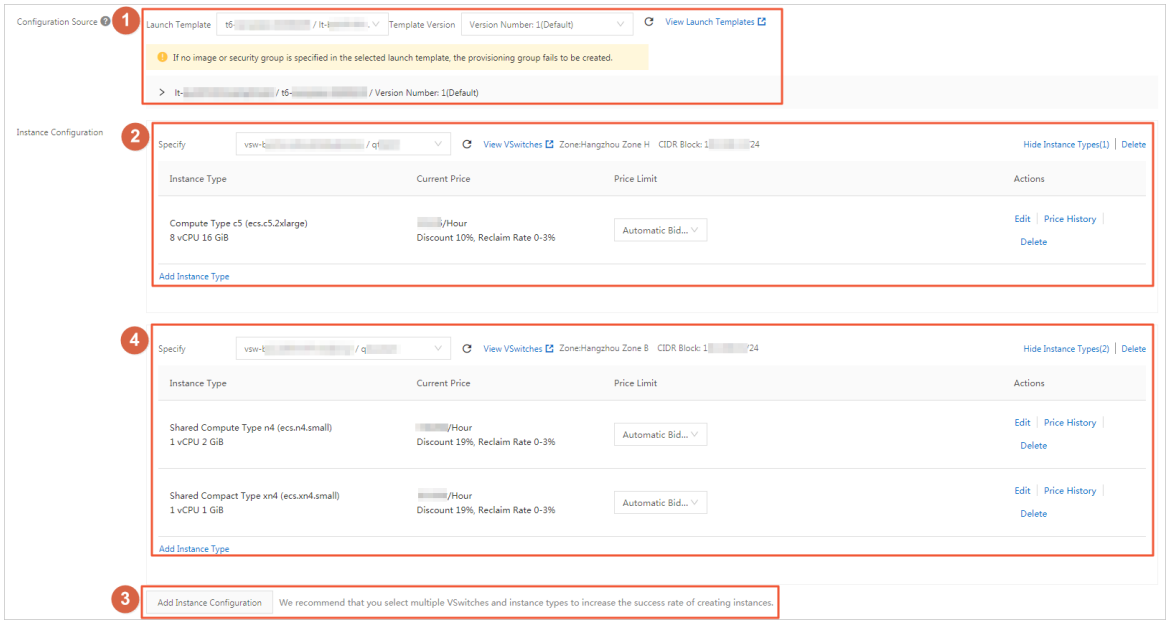
  - 12 ecs.c6.large instances
  - 8 ecs.c6.large instances and 1 ecs.c6.2xlarge instance
  - 4 ecs.c6.large instances, 2 ecs.c6.xlarge instances, and 1 ecs.c6.2xlarge instance

  > **Note** If the target capacity involves multiple instance type factors such as vCPUs and memory, call the CreateAutoProvisioningGroup operation to create an auto provisioning group and set weights for the specified instance types.

5. In the **Configuration Source** and **Instance Configuration** sections, configure instance properties.

   You can use an auto provisioning group to create instances of multiple instance types across multiple zones. If an instance cannot be created due to insufficient resources of a specific instance type or in a specific zone, the auto provisioning group attempts to create instances of another instance type or create instances in a different zone. This can improve the success rate of creating instances.

   You can specify vSwitches in multiple zones to create instances across multiple zones, and add instance types to create instances of multiple instance types. The following figure shows an example of parameter configurations. For more information, see Configure an auto provisioning group.



Two zones and three instance types are used in the preceding example. The following table describes operations involved in the example.

No.	Procedure
①	Select a launch template and then a specific version of the launch template as the configuration source. **Note** When you create instances, the vSwitches and instance types specified in section ② and section ④ are used. However, other properties such as the image, security groups, and logon credential are obtained from the configuration source.
②	Complete instance configurations. By default, the vSwitch and instance type specified in the configuration source are used. You can select other vSwitches, and select or add other instance types. **Note** You must specify at least one instance configuration for an auto provisioning group.
③	Add an instance configuration to create instances across multiple zones.

No.	Procedure
④	Complete the instance configuration. The following parameters are required: ○ **Specify**: the vSwitch to which the instances are connected. Make sure that the vSwitch belongs to a zone different from that of the vSwitch specified in section ②. **Notice** If you specify multiple vSwitches within the same zone, only the first vSwitch takes effect. ○ **Add Instance Type**: You can select one or more instance types to increase the success rate of creating instances. In this example, two instance types are specified. In the Select Instance Type dialog box, instance types that have the same instance size or vCPU-to-memory ratio as the instance type specified in the selected configuration source are listed. You can also select other instance types.

In addition to vSwitches and instance types, you can also set the maximum hourly prices for preemptible instances of each instance type. You can use one of the following methods to set the maximum hourly prices:

- Select Automatic Bidding. The real-time market price is used as the maximum hourly price for the bids. This way, preemptible instances do not fail to be created due to low bids. The instance costs are subject to changes to the market price.
- Choose Set Maximum Price > Maximum Price, and set a maximum hourly price. If you select this option, preemptible instances cannot be created if your bid is lower than the market price. This prevents instance costs from spiraling out of control if the market price increases.
- Choose Set Maximum Price > Pay-as-you-go Price, and set a percentage of the pay-as-you-go price to obtain the maximum hourly price. This costs you less than pay-as-you-go prices. For example, if you set the percentage to 50%, preemptible instances fail to be created when the market price is higher than 50% of the pay-as-you-go price.

Note We recommend that you learn the market price trends of preemptible instances before you set the maximum hourly prices. This way, you can prevent high costs or failures to meet the target capacity. You can click **Price History** in the **Actions** column to view historical prices.

6. In the **Provisioning Policy** section, select the policy used to create instances.

   The following table describes the options for this parameter.

Option	Description
Capacity Optimization Policy	The auto provisioning group selects the most cost-effective instance type based on the prices and reclaim rates to create preemptible instances. **Note** Preemptible instances may be reclaimed due to factors such as price and inventory. Instance types that have low reclaim rates are preferred.

Option	Description
Cost Optimization Policy	The auto provisioning group attempts to create ECS instances based on the unit prices of vCPUs in ascending order.
Balanced Distribution Policy	The auto provisioning group evenly distributes ECS instances across the zones that are specified in the instance configurations. This parameter takes effect only when multiple zones are specified. The preemptible instances of each instance type are reclaimed together. Instance resources are shared within the same instance family. For example, if an instance of the ecs.c6.large instance type fails to be created, a possible cause is that instance resources in the c6 instance family are insufficient. Instances of other c6 instance types such as ecs.c6.xlarge may also fail to be created. **Note** If you select Balanced Distribution Policy, we recommend that you configure instance types from different instance families to prevent instances from being reclaimed at the same time and ensure that your instance clusters remain available.

7. Configure advanced options.

   The following table describes the parameters in the Advanced section.

Parameter	Description
Group Type	○ **One-time Delivery**: After the auto provisioning group is started, it tries only once to create an instance cluster that has the target capacity. If the instance cluster fails to be created, the auto provisioning group does not try again to create the instance cluster. ○ **Continuous Delivery and Maintain Capacity**: After the auto provisioning group is started, it attempts to create an instance cluster until the cluster reaches the target capacity. If the real-time capacity does not match the target capacity, the auto provisioning group scales in or out to meet the target capacity.
Start Time	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 End Time is the validity period of the auto provisioning group. ○ **Now**: The auto provisioning group is immediately started after it is created. ○ **Specify Start Time**: Specify a point in time at which to start the auto provisioning group.

Parameter	Description
End Time	The time when the auto provisioning group expires. The period of time between this point in time and the point in time specified by Start Time is the validity period of the auto provisioning group. ○ **Never**: The auto provisioning group never expires and must be manually deleted. ○ **Specify End Time**: Specify the point in time at which the auto provisioning group expires.
Global Maximum Price for Preemptible Instances	The global maximum hourly price for preemptible instances created in the auto provisioning group. This parameter applies to preemptible instances of all instance types. If the global maximum hourly price is different from the maximum hourly price specified for a single instance type in the instance configurations, the lower one of the two prices takes precedence. ○ **Automatic Bidding**: The real-time market price is used as the maximum hourly price for the bids. This way, preemptible ECS instances do not fail to be created due to low bids. The instance costs are subject to changes to the market price. ○ **Set Maximum Price**: a fixed maximum hourly price. If you select this option, preemptible instances cannot be created if your bid is lower than the market price. This prevents instance costs from spiraling out of control if the market price increases.

8. Click **Create Provisioning Group**.

Result

After the auto provisioning group is created, it is started and attempts to create the instance cluster at the specified time. If Group Type is set to **Continuous Delivery and Maintain Capacity**, the auto provisioning group continuously maintains the instance cluster. The auto provisioning group attempts to create instances to meet the target capacity when preemptible instances are reclaimed, and replaces unhealthy instances in a timely manner.

Related information

- CreateAutoProvisioningGroup

5.4. Configure an auto provisioning group

This topic describes how to configure an auto provisioning group in different scenarios.

Machine learning scenarios

For example, you plan to complete a machine learning task in the next week. The task involves analyzing risk factors in mortgage loans. You have the following requirements on the instance cluster:

- Region: China (Hangzhou).
- Instances: use NVIDIA V100 GPUs. The GPU memory of a single instance does not exceed 32 GB.
- Target capacity: 20 instances.

- To minimize costs, only preemptible instances are created. The number of instances in the instance cluster can be less than the value specified by the Target Capacity parameter.
- Instances must be released after the task is completed.

The following table describes the configurations of the auto provisioning group based on the preceding requirements of the scenario.

Section	Parameter	Description
Capacity Configuration	**Target Capacity**	Configure the following settings based on the target capacity and instance category requirements: • Select **Instances** from the Target Capacity drop-down list. • Specify **20** in the spin box.
Instance Configuration	**Instance Configuration**	Perform the following operations to meet the requirements for instances that use NVIDIA V100 GPUs and the GPU memory of no larger than 32 GB per instance: 1. Select the ecs.gn6v-c8g1.2xlarge and ecs.gn6e-c12g1.3xlarge instance types from the Instance Type column. **Note** For information about instance types, see Instance family. 2. Query the amount of available resources of ecs.gn6v-c8g1.2xlarge and ecs.gn6e-c12g1.3xlarge in Hangzhou Zone H and Hangzhou Zone I. **Note** You can add instance configurations based on the zones and instance types: 1. Perform the following operations to add an instance configuration: ◦ Specify a vSwitch in Hangzhou Zone H. ◦ Add the ecs.gn6v-c8g1.2xlarge and ecs.gn6e-c12g1.3xlarge instance types. 2. Perform the following operations to add another instance configuration: ◦ Specify a vSwitch in Hangzhou Zone I. ◦ Add the ecs.gn6v-c8g1.2xlarge and ecs.gn6e-c12g1.3xlarge instance types. The following resource pools are formed after you add the preceding configurations: • The ecs.gn6v-c8g1.2xlarge instance type in Hangzhou Zone H • The ecs.gn6e-c12g1.3xlarge instance type in Hangzhou Zone H • The ecs.gn6v-c8g1.2xlarge instance type in Hangzhou Zone I • The ecs.gn6e-c12g1.3xlarge instance type in Hangzhou Zone I

Section	Parameter	Description
	Provisioning Policy	Select **Cost Optimization Policy**. After the auto provisioning group is started, the resource pool that is available at the lowest price is used to create an instance cluster.
Advanced	**Group Type**	To minimize costs, the number of instances in the instance cluster can be less than the value specified by the Target Capacity parameter. In this case, select **One-time Delivery**.
	Start Time and **End Time**	Specify the start and expiration time based on the next-week time requirement.
	Instance Shutdown Settings	• Instances in the auto provisioning group must be released after the task is completed. Therefore, select **Shut Down Instances Upon Group Expiration**. • Select **Shut Down Excessive Instances When Target Capacity Is Exceeded** to minimize costs.

Ticketing website scenarios

For example, you want to build a ticketing website to provide reliable ticketing services at all hours, especially during peak hours. You have the following requirements on the instance cluster:

- Region: China (Hangzhou).
- Instances: The number of vCPUs on a single instance does not exceed 8.
- Target capacity: 80 vCPUs.
- Minimum capacity: 60 vCPUs.
- To minimize costs, the website access experience is optimized based on the minimum computing requirements of the cluster.
- The cluster must have disaster recovery capabilities.

The following table describes the configurations of the auto provisioning group based on the preceding requirements of the scenario.

Section	Parameter	Description
Capacity Configuration	**Target Capacity**	Configure the following settings based on the target capacity and minimum capacity requirements: • Select **vCPUs** from the Target Capacity drop-down list. • Specify 80 in the spin box. • Select **Use Pay-as-you-go Instances to Provide Computing Power**.
	Pay-as-you-go Instance Capacity	Specify 60 in the spin box to meet the minimum capacity requirement.

Section	Parameter	Description
Instance Configuration	Instance Configuration	The c6 instance family is used because it is suitable for building frontend web servers. Perform the following operations to meet the requirements for instances that are equipped with no more than 8 vCPUs per instance: 1. Select ecs.c6.large, ecs.c6.xlarge, and ecs.c6.2xlarge from the Instance Type column. Note For information about instance types, see Instance family. 2. Query the amount of available resources of ecs.c6.large, ecs.c6.xlarge, and ecs.c6.2xlarge in Hangzhou Zone H, Hangzhou Zone I, and Hangzhou Zone J. Note You can add instance configurations based on the zones and instance types: 1. Perform the following operations to add an instance configuration: ○ Specify a vSwitch in Hangzhou Zone H. ○ Add the ecs.c6.large, ecs.c6.xlarge, and ecs.c6.2xlarge instance types. 2. Perform the following operations to add an instance configuration: ○ Specify a vSwitch in Hangzhou Zone I. ○ Add the ecs.c6.large, ecs.c6.xlarge, and ecs.c6.2xlarge instance types. 3. Perform the following operations to add another instance configuration: ○ Specify a vSwitch in Hangzhou Zone J. ○ Add the ecs.c6.large, ecs.c6.xlarge, and ecs.c6.2xlarge instance types. The following resource pools are formed after you add the preceding configurations: • The ecs.c6.large instance type in Hangzhou Zone H • The ecs.c6.xlarge instance type in Hangzhou Zone H • The ecs.c6.2xlarge instance type in Hangzhou Zone H • The ecs.c6.large instance type in Hangzhou Zone I • The ecs.c6.xlarge instance type in Hangzhou Zone I • The ecs.c6.2xlarge instance type in Hangzhou Zone I • The ecs.c6.large instance type in Hangzhou Zone J • The ecs.c6.xlarge instance type in Hangzhou Zone J • The ecs.c6.2xlarge instance type in Hangzhou Zone J

Section	Parameter	Description
	Provisioning Policy	Select **Balanced Distribution Policy**. After the auto provisioning group is started, it attempts to evenly create instances across zones to prevent instance creation failures caused by insufficient resources within a single zone. This can improve the disaster recovery capabilities of applications.
Advanced	**Group Type**	Select **Continuous Delivery and Maintain Capacity** to continuously provide ticketing service.
	Start Time and **End Time**	The auto provisioning group immediately starts and can be indefinitely retained to continuously provide ticketing service.
	Instance Shutdown Settings	Select **Shut Down Excessive Instances When Target Capacity Is Exceeded** to minimize costs.

The target capacity is specified in the number of vCPUs. Therefore, the weight of each instance is related to the number of vCPUs of each instance type. The following table describes the weighted price of each instance type.

> **Note** Prices in the following table are for reference only. The actual prices displayed on the buy page prevail.

Instance type	vCPU	Pay-as-you-go price (USD)	Weight	Weighted price (USD)
ecs.c6.large	2	0.06/hour	2	0.03/hour
ecs.c6.xlarge	4	0.121/hour	4	0.03025/hour
ecs.c6.2xlarge	8	0.241/hour	8	0.030125/hour

When an auto provisioning group attempts to create an instance cluster, the auto provisioning group first attempts to implement the balanced distribution policy and evenly creates instances across zones. Then, the auto provisioning group attempts to choose instance types that have lower weighted prices to create instances. If the weighted prices of all instance types are the same, the auto provisioning group chooses instance types at random to create instances.

Use multiple resource pools that have the lowest prices

If you want to minimize costs and alleviate the impact of reclaiming preemptible instances created by using a single resource pool, you can configure an auto provisioning group to use multiple resource pools that have the lowest prices to create instances.

You can configure an auto provisioning group to use multiple resource pools only when you call the CreateAutoProvisioningGroup operation to create the auto provisioning group. Make sure that you set the SpotAllocationStrategy parameter to *lowest-price* and specify the SpotInstancePoolsToUseCount parameter. For example, you can make the configurations described in the following table to create an auto provisioning group with the following results:

- Obtain five resource pools based on the lt-bp1ivgo4p5now3px**** launch template.

- Set the target capacity to 30 instances and create only preemptible instances.
- Use the three resource pools that have the lowest prices and create 10 preemptible instances from each resource pool.

Parameter	Value	Description
TotalTargetCapacity	30	Sets the target capacity to 30 instances.
SpotTargetCapacity	30	Creates 30 preemptible instances.
PayAsYouGoTargetCapacity	0	Creates no pay-as-you-go instances.
SpotAllocationStrategy	lowest-price	Uses the cost optimization policy for preemptible instances.
PayAsYouGoAllocationStrategy	lowest-price	Uses the cost optimization policy for pay-as-you-go instances.
SpotInstancePoolsToUseCount	3	Uses the three resource pools that have the lowest prices.
LaunchTemplateId	lt-bp1ivgo4p5now3px****	The ID of the launch template used to create instances.
LaunchTemplateConfig.1.VSwitchId	vsw-bp1ygryo03m39xhsy****	The ID of the vSwitch of the extended configuration 1 (Resource Pool 1).
LaunchTemplateConfig.1.InstanceType	ecs.c6e.large	The instance type of the extended configuration 1 (Resource Pool 1).
LaunchTemplateConfig.1.WeightedCapacity	1	The weight of the instance type of the extended configuration 1 (Resource Pool 1).
LaunchTemplateConfig.2.VSwitchId	vsw-bp16hgf8f3kvtcbyu****	The ID of the vSwitch of the extended configuration 2 (Resource Pool 2).
LaunchTemplateConfig.2.InstanceType	ecs.c6e.xlarge	The instance type of the extended configuration 2 (Resource Pool 2).
LaunchTemplateConfig.2.WeightedCapacity	1	The weight of the instance type of the extended configuration 2 (Resource Pool 2).
LaunchTemplateConfig.3.VSwitchId	vsw-bp1oeawdo9tj2gvjp****	The ID of the vSwitch of the extended configuration 3 (Resource Pool 3).

Parameter	Value	Description
LaunchTemplateConfig.3.InstanceType	ecs.c6e.2xlarge	The instance type of the extended configuration 3 (Resource Pool 3).
LaunchTemplateConfig.3.WeightedCapacity	1	The weight of the instance type of the extended configuration 3 (Resource Pool 3).
LaunchTemplateConfig.4.VSwitchId	vsw-bp1oeawdo9tj2gvjp****	The ID of the vSwitch of the extended configuration 4 (Resource Pool 4).
LaunchTemplateConfig.4.InstanceType	ecs.g6e.xlarge	The instance type of the extended configuration 4 (Resource Pool 4).
LaunchTemplateConfig.4.WeightedCapacity	1	The weight of the instance type of the extended configuration 4 (Resource Pool 4).
LaunchTemplateConfig.5.VSwitchId	vsw-bp1oeawdo9tj2gvjp****	The ID of the vSwitch of the extended configuration 5 (Resource Pool 5).
LaunchTemplateConfig.5.InstanceType	ecs.g6e.2xlarge	The instance type of the extended configuration 5 (Resource Pool 5).
LaunchTemplateConfig.5.WeightedCapacity	1	The weight of the instance type of the extended configuration 5 (Resource Pool 5).

Use a specific resource pool

You can use a specific resource pool to create instances. For example, if you purchase zone-level reserved instances to create ecs.c6e.large instances and to offset the bills of the instances, you can configure an auto provisioning group to create pay-as-you-go instances of the ecs.c6e.large instance type by using a specific resource pool.

You can configure an auto provisioning group to use a specific resource pool only when you call the CreateAutoProvisioningGroup operation to create the auto provisioning group. Make sure that you set the PayAsYouGoAllocationStrategy parameter to *prioritized* and set the priority of the specified resource pool to 0 (the highest priority). For example, you can make the configurations described in the following table to create an auto provisioning group with the following results:

- Obtain five resource pools based on the lt-bp1ivgo4p5now3px**** launch template.
- Set the target capacity to 20 instances, including 10 preemptible instances and 10 pay-as-you-go instances.
- Use the resource pool corresponding to the ecs.c6e.large instance type to create pay-as-you-go instances by specifying the LaunchTemplateConfig.1 parameter.

Parameter	Value	Description
TotalTargetCapacity	20	Sets the target capacity to 20 instances.
SpotTargetCapacity	10	Creates 10 preemptible instances.
PayAsYouGoTargetCapacity	10	Creates 10 pay-as-you-go instances.
SpotAllocationStrategy	lowest-price	Uses the cost optimization policy for preemptible instances.
PayAsYouGoAllocationStrategy	prioritized	Uses the cost optimization policy for pay-as-you-go instances.
LaunchTemplateId	lt-bp1ivgo4p5now3px****	The ID of the launch template used to create instances.
LaunchTemplateConfig.1.VSwitchId	vsw-bp1ygryo03m39xhsy****	The ID of the vSwitch of the extended configuration 1 (Resource Pool 1).
LaunchTemplateConfig.1.InstanceType	ecs.c6e.large	The instance type of the extended configuration 1 (Resource Pool 1).
LaunchTemplateConfig.1.WeightedCapacity	1	The weight of the instance type of the extended configuration 1 (Resource Pool 1).
LaunchTemplateConfig.1.Priority	0	The priority of the extended configuration 1. The value 0 indicates the highest priority.
LaunchTemplateConfig.2.VSwitchId	vsw-bp16hgf8f3kvtcbyu****	The ID of the vSwitch of the extended configuration 2 (Resource Pool 2).
LaunchTemplateConfig.2.InstanceType	ecs.c6e.xlarge	The instance type of the extended configuration 2 (Resource Pool 2).
LaunchTemplateConfig.2.WeightedCapacity	1	The weight of the instance type of the extended configuration 2 (Resource Pool 2).
LaunchTemplateConfig.3.VSwitchId	vsw-bp1oeawdo9tj2gvjp****	The ID of the vSwitch of the extended configuration 3 (Resource Pool 3).
LaunchTemplateConfig.3.InstanceType	ecs.c6e.2xlarge	The instance type of the extended configuration 3 (Resource Pool 3).

Parameter	Value	Description
LaunchTemplateConfig.3.WeightedCapacity	1	The weight of the instance type of the extended configuration 3 (Resource Pool 3).
LaunchTemplateConfig.4.VSwitchId	vsw-bp1oeawdo9tj2gvjp****	The ID of the vSwitch of the extended configuration 4 (Resource Pool 4).
LaunchTemplateConfig.4.InstanceType	ecs.g6e.xlarge	The instance type of the extended configuration 4 (Resource Pool 4).
LaunchTemplateConfig.4.WeightedCapacity	1	The weight of the instance type of the extended configuration 4 (Resource Pool 4).
LaunchTemplateConfig.5.VSwitchId	vsw-bp1oeawdo9tj2gvjp****	The ID of the vSwitch of the extended configuration 5 (Resource Pool 5).
LaunchTemplateConfig.5.InstanceType	ecs.g6e.2xlarge	The instance type of the extended configuration 5 (Resource Pool 5).
LaunchTemplateConfig.5.WeightedCapacity	1	The weight of the instance type of the extended configuration 5 (Resource Pool 5).

5.5. View an auto provisioning group

This topic describes how to view the settings of an auto provisioning group, the information of created instances in the auto provisioning group, and the execution status of scheduling tasks.

Procedure

1.
2.
3.
4. Find the auto provisioning group that you want to view and click **View Details** in the **Actions** column.

   The following table describes the information that is displayed in the **Auto Provisioning Group Details** panel.

Tab	Description

Tab	Description
Group Configurations	○ **Basic Information**: the basic information of the auto provisioning group, including the ID, type, status, start time, and expiration time of the group. ○ **Group Capacity**: the real-time and target capacities of the auto provisioning group, including the total real-time capacity, total target capacity, real-time capacity of preemptible instances, target capacity of preemptible instances, real-time capacity of pay-as-you-go instances, and target capacity of pay-as-you-go instances. ○ **Capacity-related Settings**: the capacity-related settings of the auto provisioning group, including the provisioning policies, shutdown settings for group expiration or target capacity exceeded.
Template Configurations	○ **Original Configurations**: the source configuration that includes information such as the ID and version of the instance launch template. ○ **Extended Configurations**: the extended instance configurations that include information such as instance types and zones configured for the auto provisioning group.
Instances	The list of instances that have been created in the auto provisioning group.
Group History	The scheduling tasks that the auto provisioning group has executed. If success rate of the tasks is low, check whether the settings of the auto provisioning group such as available resource pools or maximum prices are properly configured.

5.6. Modify an auto provisioning group

This topic describes how to modify the configurations of an auto provisioning group. You can modify the name, target capacity, and some capacity-related settings of an auto provisioning group.

Procedure

1.
2.
3.
4. Find the auto provisioning group that you want to modify and click **Modify** in the **Actions** column.

   The following table lists the parameters that can be modified.

Section	Description
Basic Information	The auto provisioning group name can be modified.

Section	Description
Group Capacity	The following capacities can be modified: ○ **Target Capacity**: the capacity that the auto provisioning group is scheduled to provision. You can specify this capacity in terms of **Instances** or **vCPUs**. After you select **Use Pay-as-you-go Instances to Provide Computing Power**, you can specify the computing power provided by the pay-as-you-go instances. ○ **Target Capacity of Pay-As-You-Go Instances**: the target capacity of pay-as-you-go instances that the auto provisioning group is scheduled to provision. You can specify this capacity in terms of instances or vCPUs. You can use pay-as-you-go instances to ensure that the lowest computing power requirement can be met when preemptible instances are reclaimed. **Note** When you create an auto provisioning group by calling the CreateAutoProvisioningGroup operation, you can specify the target capacity in terms of instances, vCPUs, and the memory size.
Capacity-related Settings	The following capacity-related configurations can be modified: ○ **Default Billing Method of Supplemental Instances**: **Preemptible Instances** and **Pay-As-You-Go Instances** are supported. When the sum of the Target Capacity of Pay-As-You-Go Instances and Target Capacity of Preemptible Instances values is less than the Target Capacity value, the auto provisioning group automatically creates instances that use the specified billing method to fulfill the target capacity. ○ **Global Maximum Price for Preemptible Instances**: You can modify the fixed maximum hourly price. If the specified maximum hourly price is less than the market price, preemptible instances fail to be created and high instance costs are prevented.

Note Regardless of the auto provisioning group type, a scheduling task is triggered after you modify the capacity or capacity-related settings.

5. Click **OK**.

Related information

- ModifyAutoProvisioningGroup

5.7. Delete auto provisioning groups

This topic describes how to delete auto provisioning groups. Auto provisioning groups can alleviate the instability caused by preemptible instances being reclaimed and eliminate the need to pay close attention to the status of preemptible instances.

Procedure

1.
2.

3.
4. Delete auto provisioning groups by using one of the following methods:
   - To delete a single provisioning group, find the auto provisioning group and click **Delete** in the **Actions** column.
   - To delete one or more auto provisioning groups at a time, select the auto provisioning groups and click **Delete Group** in the lower part of the page.
5. In the **Delete Group** dialog box, specify whether to delete instances in the auto provisioning group.
   - If you turn on **Delete Instances**, all instances in the auto provisioning group are released after the group is deleted.
   - If you turn off **Delete Instances**, only the auto provisioning group is deleted. All instances in the auto provisioning group are retained.
6. Click **OK**.

6.Terraform

6.1. Terraform overview

Terraform is an open source tool provided by HashiCorp that allows you to preview, configure, and manage cloud infrastructure and resources in a secure and efficient manner.

Introduction

Terraform is a tool that automates IT infrastructure orchestration. HashiCorp Terraform can use code to manage and maintain IT resources. Terraform allows you to define infrastructure resources such as virtual machines (VMs), storage accounts, and network interfaces in the configuration files that describe cloud resource topologies. The CLI of Terraform provides a simple mechanism to deploy configuration files in Alibaba Cloud or other clouds and version the configuration files. For more information, visit HashiCorp Terraform.

Terraform is a scalable tool that relies on plug-ins called Providers to support new infrastructure. Terraform allows you to use a simple template language to define, preview, and deploy cloud infrastructure in Alibaba Cloud. You can use Terraform to create, modify, and delete cloud resources, such as Elastic Compute Service (ECS), Virtual Private Cloud (VPC), ApsaraDB RDS, and Server Load Balancer (SLB) resources. For more information, see Configuration Syntax.

Alibaba Cloud is the first cloud service provider in China to integrate Terraform with its services. The Alibaba Cloud Provider offers 163 resources and 113 data sources and covers 35 Alibaba Cloud services, including computing, storage, networking, load balancing, Content Distribution Network (CDN), container, middleware, access control, and database services. This Provider is more than capable of meeting the automated cloud deployment requirements of most big customers. For more information, see Alibaba Cloud Provider.

Starting with Terraform 0.12.2, Alibaba Cloud Object Storage Service (OSS) is available as a standard remote state backend that stores state in buckets to improve state security and collaboration efficiency. For more information, see Remote State Backend.

Alibaba Cloud provides to developers a variety of modules and examples out of the box for products in categories such as computing, storage, network, middleware, and database services. Feel free to contribute your own modules. For more information, see Modules and Examples.

Benefits

- Multi-cloud infrastructure deployment

  Terraform is suitable for multi-cloud scenarios in which similar infrastructure is deployed on Alibaba Cloud, clouds of other providers, and data centers. Terraform allows developers to use the same tools and similar configuration files to manage infrastructure resources that are built on clouds of different providers.

- Automated infrastructure management

  Terraform can create configuration file templates to define and provision ECS resources in a repeatable and predictable manner. This reduces human errors during deployment and management. Terraform can deploy the same template multiple times to create identical development, test, and production environments.

- Infrastructure as code (IaC)

In Terraform, you can use code to manage and maintain resources. Terraform stores a copy of the current state of your infrastructure. This way, you can track changes made to components in the system (IaC) and share infrastructure configurations with other users.

- Reduced development costs

  You can use Terraform to create development and deployment environments based on your business requirements and reduce development and deployment costs. In addition, you can evaluate development costs before you make changes to your system.

Scenarios

For information about the scenarios of Terraform, visit the IaC - Terraform Solution page.

Use of Terraform

Terraform allows you to use a simple template language to define, preview, and deploy cloud infrastructure in Alibaba Cloud. For more information, see Configuration Syntax. Before you use Terraform to provision resources in ECS, perform the following steps:

1. Install and configure Terraform. For more information, see Use Terraform in Cloud Shell or Install and configure Terraform on your computer.
2. Use Terraform to create one or more ECS instances. For more information, see Create an ECS instance and Batch create ECS instances.
3. (Optional) Use Terraform to deploy a web cluster. For more information, see Deploy a web cluster.

For more information about how to use Terraform, visit What is Terraform?

References

- Terraform Alibaba provider
- Terrafrom Alibaba github
- Terraform Registry Alibaba Modules

6.2. Use Terraform in Cloud Shell

Alibaba Cloud Cloud Shell is a free O&M product that comes pre-installed with Terraform and configured with authentication credentials. Therefore, you can run Terraform commands in Cloud Shell.

To use Terraform in Cloud Shell, perform the following operations:

1. Open your browser and enter https://shell.aliyun.com in the address bar to access Cloud Shell.

   For more information about how to use Cloud Shell, see Use Cloud Shell.
2. Log on to Cloud Shell.
3. Compile a Terraform template.

   You can use the vim command to compile a Terraform template. If Object Storage Service (OSS) has been activated, you can upload the template that you compile to a bucket created for Cloud Shell.

   Example:

```
provider "alicloud" {}
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-hangzhou-b"
}
resource "alicloud_security_group" "default" {
  name = "default"
  vpc_id = alicloud_vpc.vpc.id
}
resource "alicloud_instance" "instance" {
  # Zone
  availability_zone = "cn-hangzhou-b"
  # Security group
  security_groups = alicloud_security_group.default. *.id
  # Instance type
  instance_type        = "ecs.n2.small"
  # System disk category
  system_disk_category = "cloud_efficiency"
  # System image
  image_id             = "ubuntu_140405_64_40G_cloudinit_20161115.vhd"
  # Instance name
  instance_name        = "test_foo"
  # VSwitch
  vswitch_id = alicloud_vswitch.vsw.id
  # Public bandwidth
  internet_max_bandwidth_out = 10
}
resource "alicloud_security_group_rule" "allow_all_tcp" {
  type              = "ingress"
  ip_protocol       = "tcp"
  nic_type          = "intranet"
  policy            = "accept"
  port_range        = "1/65535"
  priority          = 1
  security_group_id = alicloud_security_group.default.id
  cidr_ip           = "0.0.0.0/0"
}
```

4. Run the **terraform init** command to initialize the Terraform configuration files.
5. Run the **terraform plan** command to preview configurations.

```
shell@Alicloud:~$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  + alicloud_instance.master[0]
      id:                          <computed>
      availability_zone:           "cn-beijing-f"
      host_name:                   "sample"
      image_id:                    "centos_7_04_64_20G_alibase_201701015.vhd"
      instance_charge_type:        "PostPaid"
      instance_name:               "instance_sample"
      instance_type:               "ecs.t5-lc1m1.small"
      internet_charge_type:        "PayByTraffic"
      internet_max_bandwidth_in:   <computed>
      internet_max_bandwidth_out:  "1"
      key_name:                    <computed>
      password:                    <sensitive>
      private_ip:                  <computed>
      public_ip:                   <computed>
      role_name:                   <computed>
      security_groups.#:           <computed>
      spot_strategy:               "NoSpot"
      status:                      <computed>
      subnet_id:                   <computed>
      system_disk_category:        "cloud_efficiency"
      system_disk_size:            "40"
      vswitch_id:                  "${alicloud_vswitch.vsw.id}"

  + alicloud_instance.master[1]
      id:                          <computed>
      availability_zone:           "cn-beijing-f"
      host_name:                   "sample"
      image_id:                    "centos_7_04_64_20G_alibase_201701015.vhd"
      instance_charge_type:        "PostPaid"
```

6. Run the **terraform apply** command to create one or more ECS instances.

6.3. Install and configure Terraform on your computer

Before you can use the simple template language of Terraform to define, preview, and deploy cloud infrastructure, you must install and configure Terraform.

Procedure

1. Download a software package suitable for your operating system from the official Terraform website.
2. Decompress the package to the */usr/local/bin* directory.

   If you want to extract the executable file into another directory when you decompress the package, you must set an environment variable for the file by using one of the following methods:

- For Linux operating systems, use the method described in How to permanently set $PATH on Linux/Unix?
- For Windows operating systems, use the method described in Where can I set path to make.exe on Windows?
- For macOS operating systems, use the method described in How to permanently set $PATH on Linux/Unix?

3. Run the `terraform` command to verify the path configurations.
   If Terraform is installed, a command output similar to the following one is returned and contains available Terraform options:

```
username:~$ terraform
Usage: terraform [-version] [-help] <command> [args]
```

4. Create a Resource Access Management (RAM) user and grant permissions to the user to ensure high flexibility and security in permission management.
   i. Log on to the RAM console.
   ii. Create a RAM user named *Terraform* and create an AccessKey pair for the user. For more information, see Create a RAM user and Obtain an AccessKey pair.
   iii. Grant permissions to the RAM user. In this example, grant the *AliyunECSFullAccess* and `AliyunVPCFullAccess` permissions to the `Terraform` RAM user. For more information, see Grant permissions to a RAM user.
5. Create an environment variable to store authentication information.

```
export ALICLOUD_ACCESS_KEY="LTAIUrZCw3********"
export ALICLOUD_SECRET_KEY="zfwwWAMWIAiooj14GQ2*************"
export ALICLOUD_REGION="cn-beijing"
```

6.4. Create an ECS instance

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

Prerequisites

Before you begin, ensure that you have completed the following operations:

- Prepare an Alibaba Cloud account and an AccessKey pair (AccessKey ID and AccessKey secret) to use Terraform. You can go to the Security Management page of the Alibaba Cloud console to create or view your AccessKey pair.
- Install and configure Terraform. For more information, see Install and configure Terraform in the local PC and Use Terraform in Cloud Shell.

Procedure

1. Create a VPC and a vSwitch.

i. Create the *terraform.tf* file, enter the following content, and save the file to the current working directory.

```
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"
}
```

ii. Run the `terraform apply` command to create the VPC and vSwitch.

iii. 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 in the VPC created in the previous step, and add a security group rule to allow access from all IP addresses.

i. In *terraform.tf*, add the following content:

```
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        = "1/65535"
  priority          = 1
  security_group_id = alicloud_security_group.default.id
  cidr_ip           = "0.0.0.0/0"
}
```

ii. Run the `terraform apply` command to create the security group and add the security group rule.

iii. 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 ECS instance.

i. In *terraform.tf*, 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_18_04_64_20G_alibase_20190624.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. Therefore, the ECS instance is assigned a public IP address automatically.
> - For a detailed description of the parameters, see Parameter description.

ii. Run the `terraform apply` command to create the ECS instance.

iii. Run the `terraform show` command to view the created ECS instance.

iv. Run the **ssh root@<publicip>** command and enter the password to access the ECS instance.

```
provider "alicloud" {}
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"
}
resource "alicloud_security_group" "default" {
  name = "default"
  vpc_id = alicloud_vpc.vpc.id
}
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_18_04_64_20G_alibase_20190624.vhd"
  instance_name        = "test_foo"
  vswitch_id = alicloud_vswitch.vsw.id
  internet_max_bandwidth_out = 10
}
resource "alicloud_security_group_rule" "allow_all_tcp" {
  type              = "ingress"
  ip_protocol       = "tcp"
  nic_type          = "intranet"
  policy            = "accept"
  port_range        = "1/65535"
  priority          = 1
  security_group_id = alicloud_security_group.default.id
  cidr_ip           = "0.0.0.0/0"
}
```

6.5. Batch create ECS instances

This topic describes how to use Terraform to batch create ECS instances.

Prerequisites

Before you begin, make sure that you have completed the following operations:

- Prepare an Alibaba Cloud account and an AccessKey pair (AccessKey ID and AccessKey secret) to use Terraform. You can go to the Security Management page of the Alibaba Cloud console to create or view your AccessKey pair.
- Install and configure Terraform. For more information, see Install and configure Terraform in the local PC and Use Terraform in Cloud Shell.

Procedure

1. Create a VPC and a vSwitch.
   i. Create the *terraform.tf* file, enter the following content, and then save the file to the current working directory.

```
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"
}
```

   ii. Run the `terraform apply` command to create the VPC and vSwitch.
   iii. 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 within the created VPC, and then add a security group rule to allow access from all IP addresses.
   i. In *terraform.tf*, add the following content:

```
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        = "1/65535"
  priority          = 1
  security_group_id = alicloud_security_group.default.id
  cidr_ip           = "0.0.0.0/0"
}
```

   ii. Run the `terraform apply` command to create a security group and add a security group rule.
   iii. 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 batch create ECS instances. In this example, three ECS instances are created.

i. In the *terraform.tf* file, add the following content:

```
module "tf-instances" {
 source                      = "alibaba/ecs-instance/alicloud"
 region                      = "cn-beijing"
 number_of_instances         = "3"
 vswitch_id                  = alicloud_vswitch.vsw.id
 group_ids                   = [alicloud_security_group.default.id]
 private_ips                 = ["172.16.0.10", "172.16.0.11", "172.16.0.12"]
 image_ids                   = ["ubuntu_18_04_64_20G_alibase_20190624.vhd"]
 instance_type               = "ecs.n2.small"
 internet_max_bandwidth_out  = 10
 associate_public_ip_address = true
 instance_name               = "my_module_instances_"
 host_name                   = "sample"
 internet_charge_type        = "PayByTraffic"
 password                    = "User@123"
 system_disk_category        = "cloud_ssd"
 data_disks = [
  {
    disk_category = "cloud_ssd"
    disk_name     = "my_module_disk"
    disk_size     = "50"
  }
 ]
}
```

Note In the preceding example, a public IP address is assigned to the instance because `associate_public_ip_address = true` and `internet_max_bandwidth_out = 10`. For a detailed description of the parameters, see Parameter description.

ii. Run the `terraform apply` command to create the ECS instance.

iii. Run the `terraform show` command to view the created ECS instance.

iv. Run the **ssh root@<publicip>** command and enter the password to access the ECS instances.

```
provider "alicloud" {}
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"
}
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        = "1/65535"
  priority          = 1
  security_group_id = alicloud_security_group.default.id
  cidr_ip           = "0.0.0.0/0"
}
module "tf-instances" {
 source                      = "alibaba/ecs-instance/alicloud"
 region                      = "cn-beijing"
 number_of_instances         = "3"
 vswitch_id                  = alicloud_vswitch.vsw.id
 group_ids                   = [alicloud_security_group.default.id]
 private_ips                 = ["172.16.0.10", "172.16.0.11", "172.16.0.12"]
 image_ids                   = ["ubuntu_18_04_64_20G_alibase_20190624.vhd"]
 instance_type               = "ecs.n2.small"
 internet_max_bandwidth_out  = 10
 associate_public_ip_address = true
 instance_name               = "my_module_instances_"
 host_name                   = "sample"
 internet_charge_type        = "PayByTraffic"
 password                    = "User@123"
 system_disk_category        = "cloud_ssd"
 data_disks = [
  {
    disk_category = "cloud_ssd"
    disk_name     = "my_module_disk"
    disk_size     = "50"
  }
 ]
}
```

Related information

- terraform-alicloud-ecs-instance

6.6. Deploy a web cluster

This topic describes how to deploy a web cluster by using Terraform.

Prerequisites

Before you begin, make sure that you have completed the following operations:

- Prepare an Alibaba Cloud account and an AccessKey pair (AccessKey ID and AccessKey secret) to use Terraform. You can go to the Security Management page of the Alibaba Cloud console to create or view your AccessKey pair.
- Install and configure Terraform. For more information, see Install and configure Terraform in the local PC and Use Terraform in Cloud Shell.

Context

Before you deploy a website or application, you must deploy a series of nodes. Server Load Balancer (SLB) distributes requests to each node and automatically scales based on the access quantity or resource usage. This example deploys the entire application in a single zone and only allows access to the Hello World page through port 8080.

Procedure

1. Create a VPC and a VSwitch.
   i. Create the *terraform.tf* file, enter the following content, and save the file to the current working directory.

```
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"
}
```

   ii. Run the `terraform apply` command to create a VPC and a VSwitch.
   iii. 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.

i. In *terraform.tf*, add the following content:

```
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        = "1/65535"
  priority          = 1
  security_group_id = alicloud_security_group.default.id
  cidr_ip           = "0.0.0.0/0"
}
```

ii. Run the `terraform apply` command to create a security group.

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

3. Create a Server Load Balancer (SLB) instance and assign a public IP address for it. This example configures a mapping from frontend port 80 to backend port 8080 for the SLB instance and displays a public IP address for subsequent tests.

i. 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
}
```

ii. Run the `terraform apply` command to create an SLB instance.

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

4. Create Auto Scaling resources.

This example creates the following resources:

- Scaling group: specifies the minimum number of instances as 2 and the maximum number of instances as 10. The created SLB instance is attached to the scaling group. The depends_on attribute specifies the deployment sequence because the scaling group depends on SLB listener configurations.
- Scaling group configuration: specifies the specific configuration of the ECS instance. The initialization configuration (user-data) generates a Hello World page and provides services through port 8080. To simplify operations, this example assigns a public IP address for the ECS instance and configures `force_delete=true` to subsequently delete the environment.
- Scaling rule: defines the specific scaling rule.

  i. Create the *ess.tf* file and add the following content:

```
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_18_04_64_20G_alibase_20190624.vhd"
  instance_type = "ecs.c5.large"
  security_group_id = alicloud_security_group.default.id
  active= true
  enable= true
  user_data = "#! /bin/bash\necho \"Hello, World\" > index.html\nnohup busybox http
d -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
}
```

  ii. Run the `terraform apply` command to create resources.

  After the resources are created, the public IP address of the SLB instance is displayed.

  Auto Scaling will create an ECS instance after two minutes.

  iii. Run the `curl http://<slb public ip>` command to verify the results.

  If `Hello, World` is displayed, it indicates that you can use the SLB instance to access the webpage provided by the ECS instance.

5. Run the `terraform destroy` command to delete the test environment. After you confirm, the

entire 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.