ALIBABA CLOUD

Alibaba Cloud

Elastic Compute Service

Elasticity

Document Version: 20200903

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 05
1.1. Launch templates 05
1.2. Create a launch template 05
2.Create a scaling group based on an existing ECS instance 08
3.Resource Orchestration Service 16
3.1. ROS overview 16
4.Deployment sets 18
4.1. Overview 18
4.2. Create a deployment set 19
4.3. Create an ECS instance in a deployment set 20
4.4. Change the deployment set of an instance 22
4.5. Delete a deployment set 24
5.Manage auto provisioning groups 25
5.1. Auto Provisioning overview 25
5.2. Configure an auto provisioning group 27
5.3. Create an auto provisioning group 30
5.4. View an auto provisioning group 34
5.5. Modify an auto provisioning group 35
5.6. Delete auto provisioning groups 36
6.Terraform 37
6.1. Install and configure Terraform 37
6.2. Create an ECS instance 37
6.3. Create multiple ECS instances 41
6.4. Deploy a web cluster 45

1.Launch template

1.1. Launch templates

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

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

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

Console operations

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

API operations

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

1.2. Create a launch template

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

Context

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

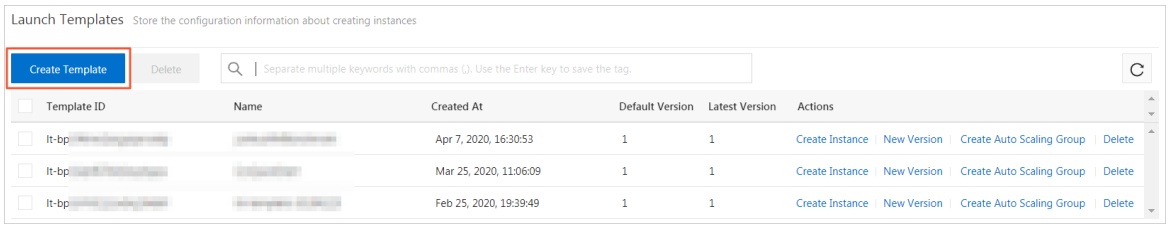
- You can create up to 30 launch templates in each region.
- When you create a launch template, all parameters are optional. However, if a launch template does not contain required parameters such as the instance type or image, you must specify these parameters when you create an instance by using the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

Create a launch template from the ECS console

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

1. Log on to the ECS console.

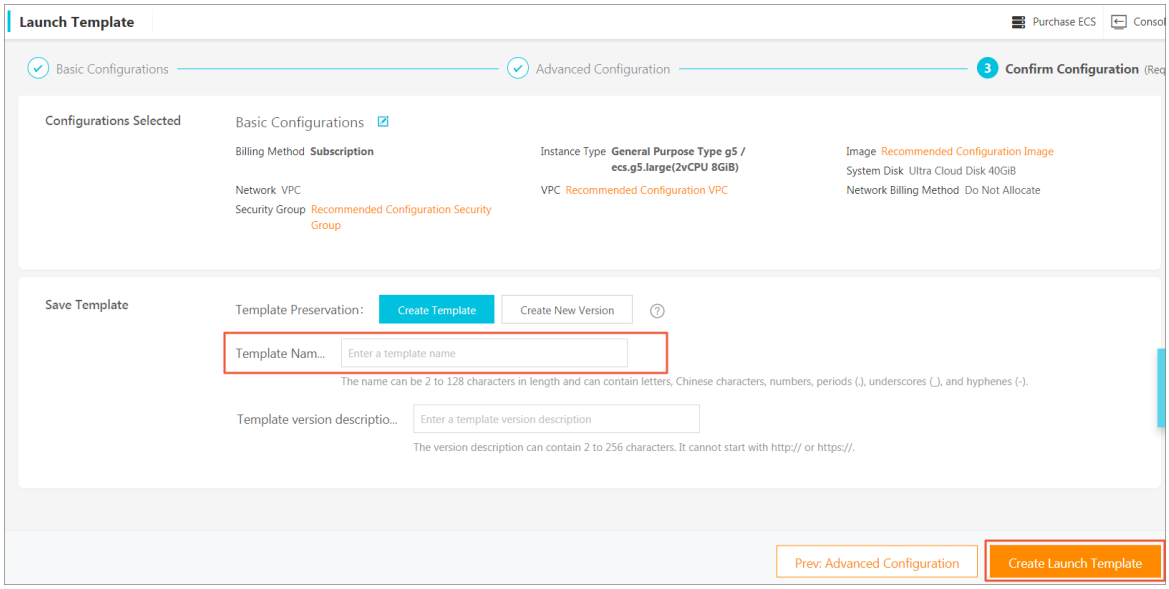
2. In the left-side navigation pane, choose **Deployment & Elasticity > Launch Templates**.
3. In the top navigation bar, select a region.
4. On the Launch Templates page, click **Create Template**.



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

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

6. In the **Confirm Configuration** step, enter a template name and a template version description. Click **Create Launch Template**.



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

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

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

Create a launch template on the ECS buy page

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

1. Go to the Elastic Compute Service product homepage. Click **Buy Now**.

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

Related information

- CreateLaunchTemplate

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

This topic describes how to create a scaling group based on an existing ECS instance.

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

Prerequisites

- Before you associate a scaling group with SLB instances, make sure that the following conditions are met:
  - You have at least one SLB instance in the **Running** state. For more information, see Create an SLB instance.
  - The SLB instance and the scaling group are in the same region.
  - The SLB instance and the scaling group are in the same Virtual Private Cloud (VPC) network if their network type is VPC.
  - If the network type of the SLB instance is classic network, the network type of the scaling group is VPC, and the back-end server group of the SLB instance contains VPC-connected ECS instances, then the ECS instances and the scaling group must be in the same VPC network.
  - At least one listener is configured on the SLB instance. For more information, see Listener overview.
  - Health check is enabled on the SLB instance. For more information, see Configure health check.
- Before you associate a scaling group with ApsaraDB for RDS instances, make sure that the following conditions are met:
  - At least one ApsaraDB for RDS instance is in the **Running** state. For more information, see What is ApsaraDB for RDS?.
  - The ApsaraDB for RDS instances and the scaling group must reside in the same region.

Context

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

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

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

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images** > **Instances**.

3. In the top navigation bar, select a region.
4. Find the ECS instance that you want to use as a scaling configuration source, and choose **More > Deployment & Elasticity > Create Scaling Group** in the **Actions** column.
5. On the **Create Scaling Group** page, view the scaling configuration source. When you create a scaling group based on an ECS instance, the Select Existing Instance option and the ECS instance are automatically selected in the Source Type section on the Create Scaling Group page. Keep these settings unchanged.
6. Configure basic information of the scaling group.
   i. Specify the scaling group name.
   ii. Specify the number of instances in the scaling group.

Parameter	Description
Maximum Number of Instances	When the number of existing ECS instances is greater than the maximum number of instances, Auto Scaling automatically removes ECS instances from the scaling group to maintain the maximum number.
Minimum Number of Instances	When the number of existing ECS instances is less than the minimum number of instances, Auto Scaling automatically adds instances to the scaling group to maintain the minimum number.
Expected Number of Instances	Auto Scaling automatically keeps the number of ECS instances at the expected level. For more information, see Expected Instances.

   iii. Specify the default cooldown time. The Default Cooldown Time parameter specifies the default cooldown time of a scaling group in seconds. During the cooldown time, the scaling group rejects all scaling activity requests triggered by event-triggered tasks from Cloud Monitor. However, scaling activities triggered by other types of tasks, such as scheduled tasks and manually executed tasks, are not subject to the cooldown time and are executed immediately.
   iv. (Optional)Specify the instance removing policy. This policy is used to filter and remove ECS instances from a scaling group based on multiple filter conditions. If multiple ECS instances meet the conditions of the policy, one instance is removed at random.

   The Instance Removing Policy parameter contains **Filter First** and **Then Remove from Results**. You cannot specify the same value for **Filter First** and **Then Remove from Results**. The following table describes the options for the instance removing policy.

Option	Description

Option	Description
Earliest Instance Created Using Scaling Configuration	The scaling configuration refers to the instance configuration source that contains the scaling configuration and instance launch template. Auto Scaling filters instances to find the ones that use the earliest scaling configuration and instance launch template. Manually added instances are not associated with scaling configurations or instance launch templates. Therefore, manually added instances are not found first. If all associated instances have been removed but Auto Scaling must remove more instances from the scaling group, manually added instances are removed at random. The version of an instance launch template does not indicate the order in which the template was added. For example, you select the lt-foress V2 template when you create a scaling group. Then, you select the lt-foress V1 template to modify the scaling group. The scaling group considers the lt-foress V2 launch template as the template that was added earlier.
Earliest Created Instance	Auto Scaling filters instances to find the ones that were created at the earliest points in time.
Most Recent Created Instance	Auto Scaling filters instances to find the ones that were created most recently.
No Policy	This option is available only for **Then Remove from Results**. This option indicates that Auto Scaling does not filter instances in the second step.

For example, if Auto Scaling filters instances based on **Earliest Instance Created Using Scaling Configuration** in the first step, then you can select one of the following options in the second step:

- **No Policy**: This option indicates that Auto Scaling does not filter instances in the second step.
- **Earliest Created Instance**: Auto Scaling filters instances to find the ones that were created at the earliest points in time from the filtering results in the first step.
- **Most Recent Created Instance**: Auto Scaling filters the instances to find the ones that were created most recently from the filtering results in the first step.

v. (Optional)Enable deletion protection for the scaling group. After this feature is enabled, the scaling group cannot be deleted from the console or through API operations.

vi. Add tags. You can add tags to search for and manage scaling groups. For more information, see Overview.

7. Specify the scaling configuration.

i. View the network type.

> **Notice** Keep the default network type unchanged. Otherwise, the creation fails.

Network type of instances within the scaling group	Description
VPC	■ When you create a scaling configuration, you can select only instance types that support VPCs. ■ When you manually add existing ECS instances to the scaling group, you can select only instances that are located in the same VPC as the scaling group.
Classic Network	■ When you create a scaling configuration, you can select only instance types that support the classic network. ■ When you manually add existing ECS instances to the scaling group, you can select only instances that are located in the classic network.

ii. (Optional)If the network type of instances within the scaling group is **VPC**, configure the following parameters:

- Network type: VPC

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

- VSwitch

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

- Multi-zone scaling policy

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

Policy	Description
Priority Policy	The VSwitch that is first selected has the highest priority. When an ECS instance cannot be created in the zone where the VSwitch with the highest priority resides, Auto Scaling automatically uses the VSwitch that has the second highest priority to create the ECS instance.

Policy	Description
Balanced Distribution Policy	The balanced distribution policy takes effect only when the scaling group is associated with multiple VSwitches that are distributed across more than two zones. The policy evenly distributes ECS instances across zones where the VSwitches reside. If the ECS instances are not evenly distributed across zones due to insufficient resources, you can use the Rebalance Distribution feature to evenly distribute the ECS instances. For more information, see Rebalance the distribution of ECS instances.
Cost Optimization Policy	The cost optimization policy takes effect only when you specify multiple instance types in the scaling configuration. Auto Scaling creates ECS instances based on the unit prices of vCPUs in ascending order. If you select Preemptible Instance as the billing method in the scaling configuration, preemptible instances are preferentially created. When preemptible instances cannot be created due to insufficient resources, Auto Scaling automatically attempts to create pay-as-you-go instances.

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

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

- Instance reclaim mode

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

Mode	Description
Release Mode	When a scale-in event is triggered, Auto Scaling automatically releases a specific number of ECS instances. When a scale-out event is triggered, Auto Scaling automatically creates a specific number of ECS instances for the scaling group.
Shutdown and Reclaim Mode	This mode can improve scaling efficiency. When a scale-in event is triggered, the status of the removed ECS instances becomes No Fees for Stopped Instances (VPC-Connected), and their vCPUs, memory, and public IP addresses are reclaimed. You are no longer charged for these resources. However, you are still charged for other resources such as disks and elastic IP addresses (EIPs). For more information, see No Fees for Stopped Instances (VPC-Connected). These stopped ECS instances form a stopped instance pool. **Note** If the ECS instances have public IP addresses before they enter the No Fees for Stopped Instances (VPC-Connected) state, the instances are reassigned public IP addresses when they are restarted, and the addresses may be different from the previous ones. When a scale-out event is triggered, the ECS instances in the stopped instance pool preferentially enter the Running state. If these instances are still insufficient, Auto Scaling creates more ECS instances. When a scale-out event is triggered, the ECS instances in the stopped instance pool may or may not enter the Running state, depending on whether their instance types meet the requirement. If the ECS instances in the stopped instance pool cannot enter the Running state due to insuffcient qualified instances, Auto Scaling will release these instances and create a specific number of ECS instances that meet the instance type requirement to satisfy the desired capacity of the scaling group.

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

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

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

8. (Optional)Configure advanced settings. You can associate only a limited number of SLB and ApsaraDB for RDS instances with a scaling group. For more information, see Limits.

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

Server group	Port number	Weight	Description
Default server group	You can specify the port number when you configure listeners for SLB instances.	Default value: 50. You can set the weight to other values in the scaling configuration.	The group of ECS instances that are used to receive requests. If the listener is not configured with a VServer group or a primary/secondary server group, requests are forwarded to the ECS instances in the default server group.
VServer group	You can specify the port number when you select a VServer group.	Default value: 50. You can set the weight to other values when you select a VServer group.	If you want to distribute different requests to different backend servers or configure domain name- or URL-based routing methods, you can use VServer groups. You can specify only a limited number of VServer groups for a scaling group. For more information, see Limits.

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

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

9. Click **Create Scaling Group**.
10. In the **Create Scaling Group Status Wizard** message, click **View Scaling Groups**.
    The newly created scaling group is displayed in the scaling group list. Auto Scaling automatically creates a scaling configuration for the scaling group, and the scaling group enters the **Enabled** state after it is created because an existing ECS instance is used as the scaling configuration source.
    - If the value of **Minimum Number of Instances** is greater than zero, Auto Scaling automatically creates ECS instances to ensure that the number of instances in the scaling group is no less than the minimum number of instances.
    - If the value of **Expected Number of Instances** is greater than zero, Auto Scaling automatically creates or removes ECS instances to ensure that the number of instances in the scaling group is equal to the expected number of instances.

What's next

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

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

Related information

- 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 RSD instances possible.

ROS template IaC stack DevOps
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 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 ECS instances and implements disaster recovery and business availability when ECS instances are created.

Deployment policy

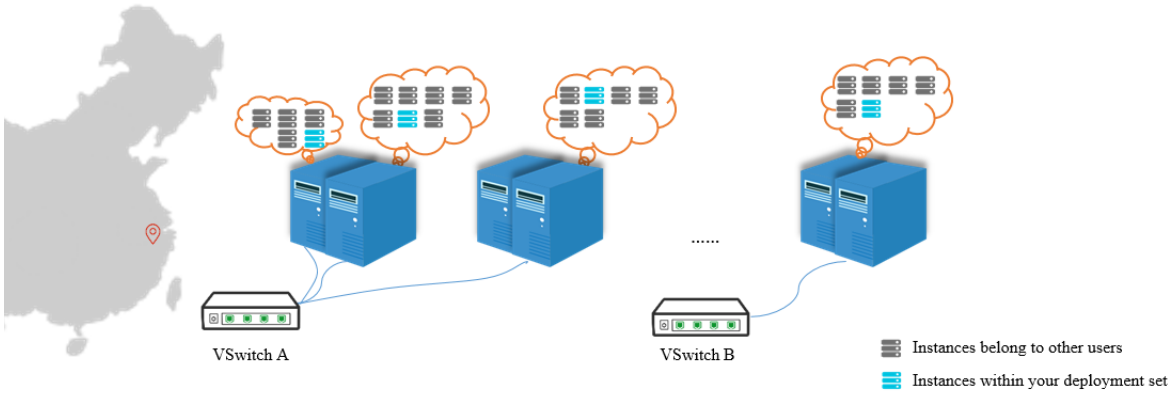
You can use a deployment set to distribute your ECS instances to different physical servers to guarantee high availability and set up underlying disaster recovery. When you create ECS instances in a deployment set, Alibaba Cloud will start the ECS instances on different physical servers within the specified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

Deployment sets support the high availability policy:

- When you use the high availability policy, all the ECS instances within your deployment set are strictly distributed across different physical servers within the specified region. The high availability policy applies to application architectures where several ECS instances must be isolated from each other. The policy significantly reduces the chances of service being unavailable.
- When you use the high availability policy, you may not be able to create an ECS instance when there is a supply shortage in the specified region. Furthermore, pay-as-you-go instances in the No Fees for Stopped Instances (VPC-Connected) state may fail to start next time. If this problem occurs, we recommend that you wait a while and then try again or create a new instance.

Deployment example

The following figure shows a typical example on how to use a deployment set to improve business reliability. In the deployment set, four ECS instances are distributed to four different physical servers.



If you want to achieve low-latency communication between ECS instances, we recommend that you make sure the network types of the instances are the same. For example, select the same VPC for the ECS instances when you create them.

Billing details

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

Limits

Before you use deployment sets, note that:

- 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 seven ECS instances in each zone. This limit varies with your ECS usage. You can use the following formula to calculate the number of ECS instances that can be created in an Alibaba Cloud region: *7 × Number of zones*.
- Instances in the following instance families can be created in deployment sets: c6, g6, r6, hfc6, hfg6, hfr6, d2, d2s, d2c, c5, d1, d1ne, g5, hfc5, hfg5, i2, i2g, i1, ic5, r5, se1ne, sn1ne, and sn2ne. For more information about instance types and their performance, see Instance families.
- Supply shortage may result in a failure to create an instance or restart a pay-as-you-go instance that is in the No Fees for Stopped Instances (VPC-Connected) state in a deployment set. For more information, see No Fees for Stopped Instances (VPC-Connected).

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
- 修改部署集信息
- 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.

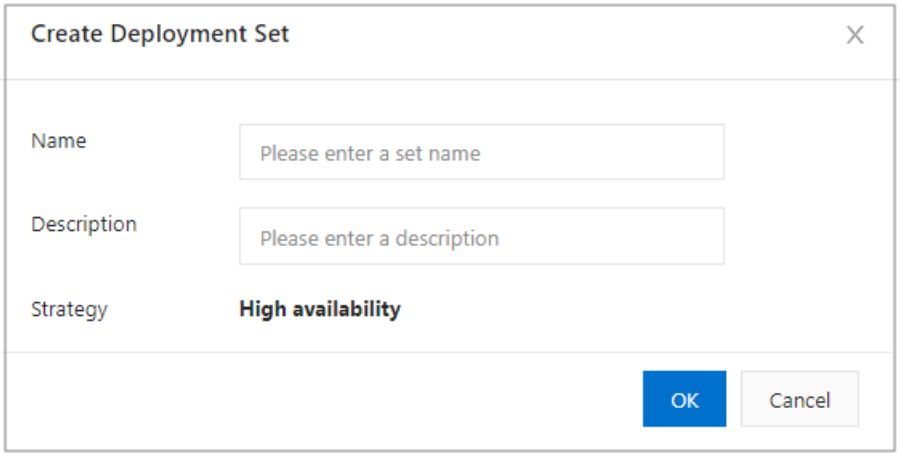
4.2. Create a deployment set

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. Log on to the ECS console.

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

Prerequisites

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

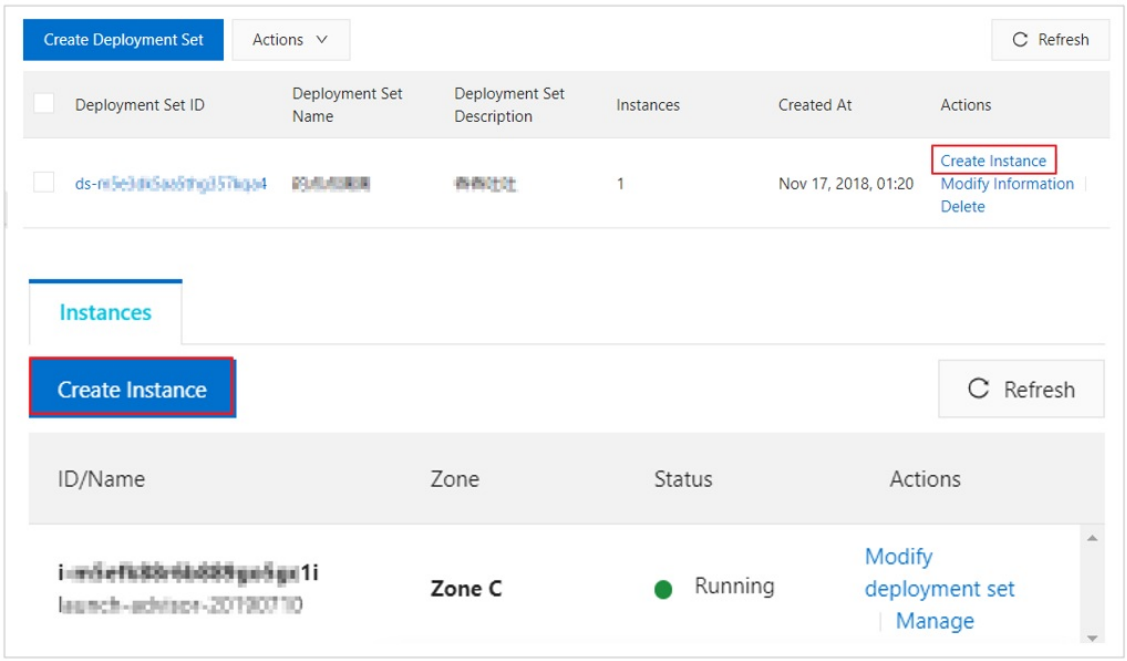
Context

When you create ECS instances in a deployment set, you can create up to seven ECS instances in each zone. You can use the following formula to calculate the maximum number of ECS instances that you can create in a deployment set within a region: 7 × Number of zones within the region. These limits vary with your ECS usage.

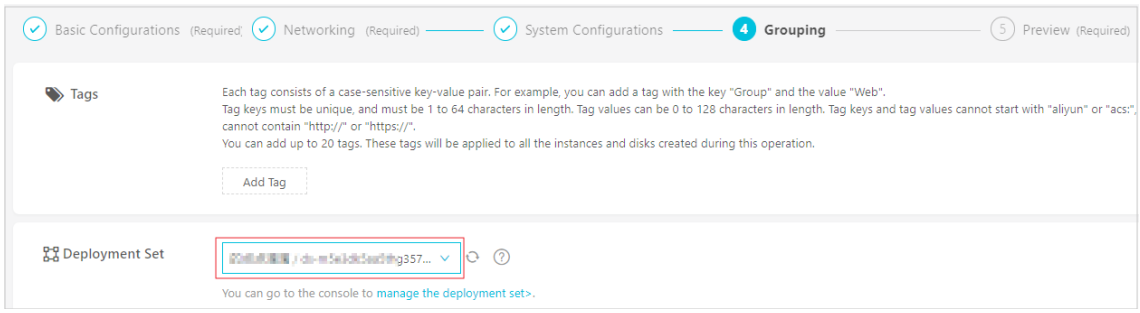
When you create an instance, you can use a launch template or use the batch creation feature to facilitate instance creation. For more information, see Launch templates.

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Deployment & Elasticity** > **Deployment Sets**.
3. In the top navigation bar, select a region.
4. On the Deployment Sets page, find the target deployment set. You can use one of the following methods to create an ECS instance in the deployment set:
   - In the **Actions** column corresponding to the deployment set, click **Create Instance**.



   - Click deployment set ID. In the **Instances** pane that appears, click **Create Instance**.
5. On the page that appears, complete the instance configuration on the **Custom Launch** tab. For more information about how to create an instance, see Create an instance by using the provided wizard. Note the following configurations when you create an instance:
   - **Basic configurations:**
     - **Region:** The ECS instance and the target deployment set must be in the same region.
     - **Zone:** Up to seven ECS instances can be created in each zone of a deployment set.
     - **Instance Type:** Instances in the following instance families can be created in deployment sets: c6, g6, r6, hfc6, hfg6, hfr6, d2, d2s, d2c, c5, d1, d1ne, g5, hfc5, hfg5, i2, i2g, i1, ic5, r5, se1ne, sn1ne, and sn2ne. For more information about instance types and their performance, see Instance families.
     - **Quantity:** optional. You can specify the number of ECS instances to create. This number must be specified based on the number of ECS instances that already exist in the current zone of the deployment set.
   - **System Configuration** > **Sequential Suffix:** optional. After you create multiple instances, you can add sequential suffixes to the instance names and hostnames. The sequential suffix ranges from 001 to 999.
   - **Grouping** > **Deployment Set:** Select the target deployment set.



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

6. Check the settings you have made, and then click **Create Order** or **Create Instance**.
7. In the left-side navigation pane, choose **Deployment & Elasticity > Deployment Sets**. On the Deployment Sets page, find the deployment set to view the instance that you have created.

What's next

After you create an ECS instance, you can perform the following operations:

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

Related information

- RunInstances

4.4. Change the deployment set of an instance

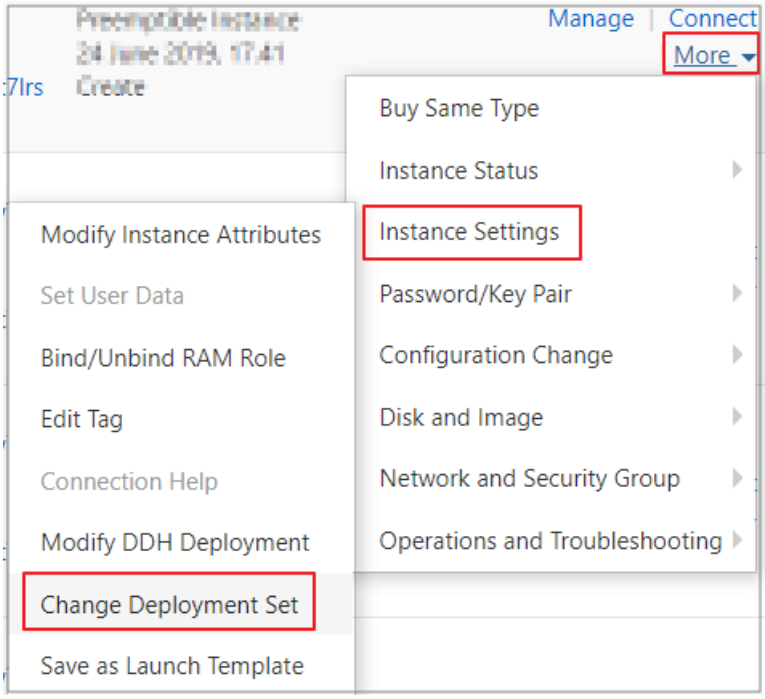
This topic describes how to change the deployment set of an instance by using the ECS console.

Prerequisites

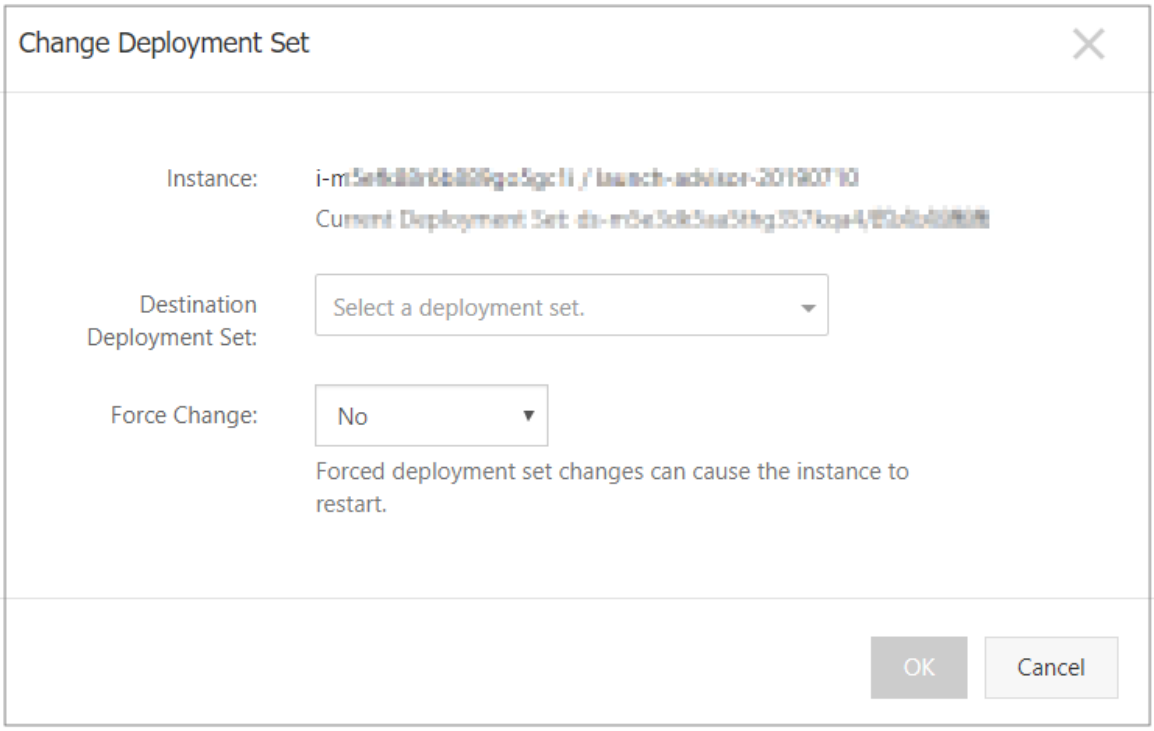
A deployment set is created for your instance. For more information, see Create a deployment set.

Procedure

1.
2.
3.
4. Find the target instance. The instance must be in the **Stopped** or **Running** state.
5. In the **Actions** column, choose **More > Instance Settings > Change Deployment Set**.



6. In the **Change Deployment Set** dialog box, select the destination deployment set and specify the **Force Change** option:
   - **Yes**: You can change the host of an instance and restart an instance that is in the **Running** or **Stopped** state.
   - **No**: You cannot change the host of an instance. The deployment set must be added on the current host. This may cause the deployment set modification to fail.



7. Click **OK**.

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

4.5. Delete a deployment set

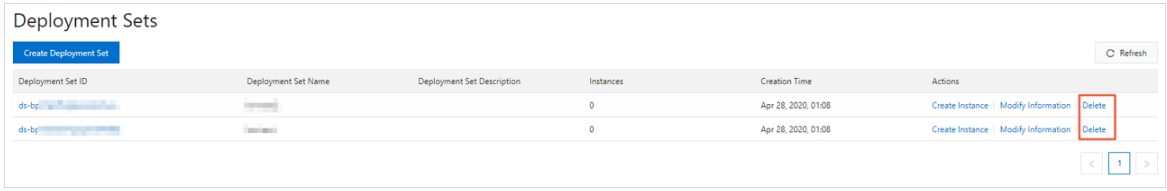
You can delete any deployment sets that you no longer need to ensure that the usage limit is not exceeded.

Prerequisites

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

Procedure

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



5. In the message that appears, click **OK**.

Related information

- DeleteDeploymentSet

5.Manage auto provisioning groups

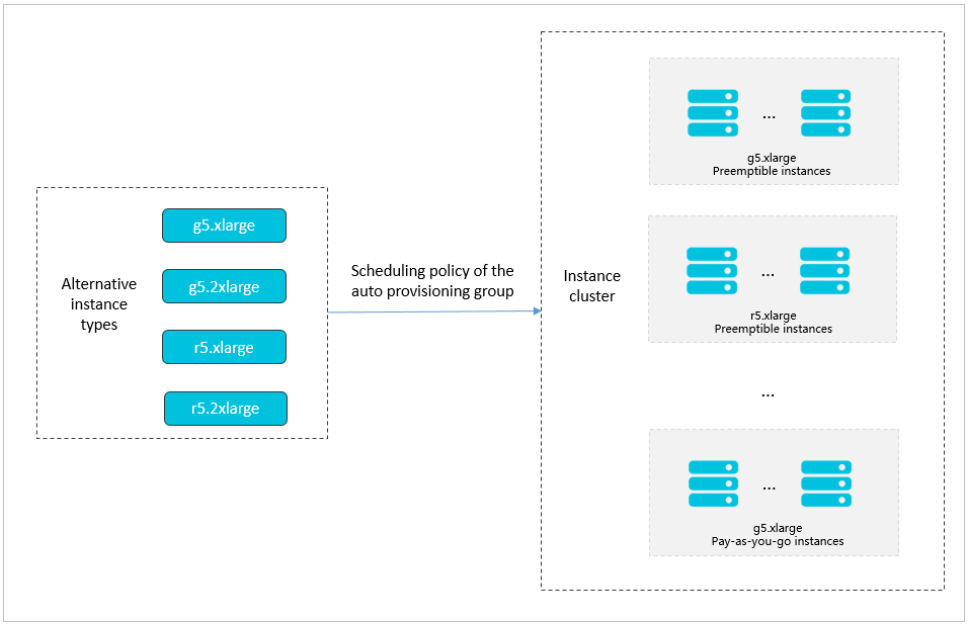
5.1. Auto Provisioning overview

Auto Provisioning is a service to quickly deploy an instance cluster that consists of preemptible and pay-as-you-go instances. Auto Provisioning supports one-click deployment of instance clusters with specified billing methods, zones, and instance families. You can use auto provisioning groups to provide stable computing power, alleviate the instability caused by the reclaiming of preemptible instances, and eliminate the need to manually create instances.

Introduction to Auto Provisioning

Auto provisioning groups can create instance clusters to meet computing power needs based on configured instances types and scheduling policies. After an auto provisioning group is created, the auto provisioning group automatically selects instance types and creates an instance cluster. You do not need to calculate the costs of instances individually.

If you select **Continuous Delivery and Maintain Capacity**, the auto provisioning group automatically compares real-time and target capacities. When preemptible instances are reclaimed, the auto provisioning group selects instance types and creates new instances to maintain the target capacity of your business at peak demand and meet your computing power needs at the lowest cost.



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

Scenarios

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

Features

- Cross billing methods

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

- Cross zones

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

- Cross instance families

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

- Flexible policy combinations

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

- Complete cost control

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

- Practical protection mechanism

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

Billing

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

> **Warning** Make sure that you have sufficient balance in your account. If you have overdue payments, all pay-as-you-go instances and preemptible instances are stopped. For more information, see Settlement period. In this case, the auto provisioning group cannot deliver new ECS instances. The group determines whether stopped instances are unhealthy based on the health check feature and then removes and releases unhealthy instances.

Limits

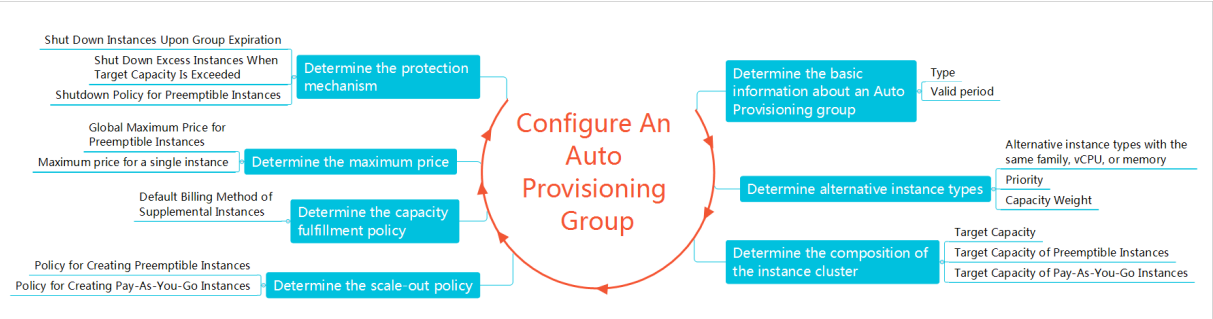
- Auto provisioning groups cannot schedule resources across regions.
- A maximum of 1,000 instances can be created in each auto provisioning group.
- Only a single launch template can be specified for each auto provisioning group. However, you can extend the launch template to implement more configurations.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

5.2. Configure an auto provisioning group

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

Procedure to configure an auto provisioning group

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



Process to deploy an instance cluster

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

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

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

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

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

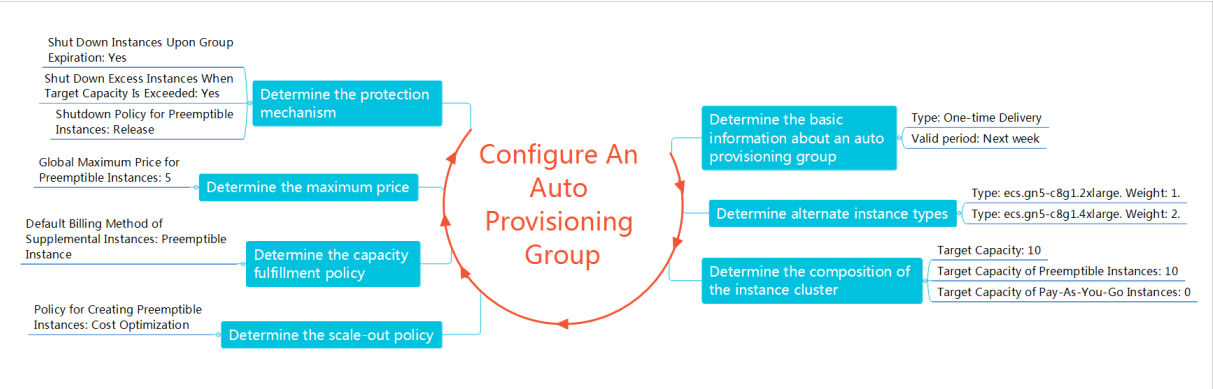
2. If the specified target capacities of preemptible and pay-as-you-go instances do not meet the target capacity requirements of the cluster, the auto provisioning group creates new instances based on **Default Billing Method of Supplemental Instances** to meet the capacity requirements.
3. If **Continuous Delivery and Maintain Capacity** is selected, the auto provisioning group continuously compares real-time and target capacities. If any of the target capacities has not been met, the auto provisioning group creates instances when resources are available to meet the target capacity.

Example of machine learning scenarios

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

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

Considering the preceding requirements, the following configurations are used.



The following two solutions can meet capacity requirements:

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

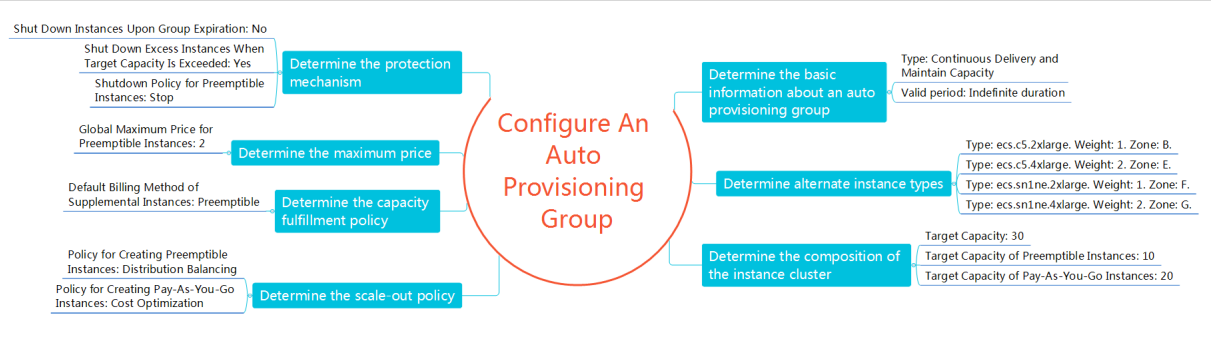
When using the Cost Optimization policy to create preemptible instances, the auto provisioning group compares the required costs of each solution for the instance cluster. The solution with the lowest cost is then selected to implement a one-time delivery to the cluster. If the actual instance cluster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

Example of ticketing website scenarios

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

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

Considering the preceding requirements, the following configurations are used.



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

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

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

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

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

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

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

5.3. Create an auto provisioning group

This topic describes how to create an auto provisioning group in the ECS console. The created group will provision an instance cluster based on your configurations.

Prerequisites

- Your account is granted permissions on Auto Provisioning.

  > **Note** When you access the Auto Provisioning page for the first time, follow the instructions to assign the `AliyunECSAutoProvisioningGroupRole` RAM role to your account.

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

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Deployment & Elasticity > Auto Provisioning**.
3. In the top navigation bar, select a region.
4. Click **Create Auto Provisioning Group**.
5. Configure the parameters in the Capacity Configuration section. These parameters determine the total capacity of the auto provisioning group and the capacity ratio of preemptible ECS instances to pay-as-you-go ECS instances. The following table describes the parameters.

Parameter	Description
Group Name	The name of the auto provisioning group. The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), periods (.), and hyphens (-). It must start with a letter and cannot start with http:// or https://.
Target Capacity	The capacity that the auto provisioning group is scheduled to provision. You can specify this capacity in terms of the number of ECS instances or vCPUs. You can select **Use Pay-as-you-go Instances to Provide Computing Power** and then specify the capacity of pay-as-you-go ECS instances to be provisioned.
Pay-as-you-go Instance Capacity	The capacity of pay-as-you-go ECS instances that the auto provisioning group is scheduled to provision. You can specify this capacity in terms of the number of ECS instances or vCPUs.

6. Configure the parameters in the Instance Configuration section. These parameters determine the attributes of the ECS instances to be provisioned, such as instance types, images, security groups, networks, and VSwitches. You must specify a launch template as the configuration source. Then, you can add one or more alternative instance configurations to increase the success rate of creating ECS instances.
   i. **Configuration Source**: Select a launch template and a template version from the drop-down lists.
   ii. **Instance Configuration**: At least one instance configuration is required in the auto provisioning group. You can click **Add Instance Configuration** to add multiple alternative instance configurations. The following table describes the parameters in the Instance Configuration section. These parameters appear only after a launch template and a template version are selected.

Parameter	Description
Specify VSwitch	The VSwitch which to connect to the ECS instances provisioned by the auto provisioning group. You can specify VSwitches from different zones to provide zone redundancy. If ECS instances fail to be created in a zone due to insufficient resources, the auto provisioning group attempts to create ECS instances in other zones. This helps increase the creation success rate.
Instance Type	The alternative instance type. You can specify multiple instance types to improve provisioning flexibility. If ECS instances of an instance type fail to be created due to insufficient resources, the auto provisioning group attempts to create ECS instances of other instance types. This helps increase the creation success rate. You can click **Add Instance Type** to add alternative instance types. 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 provided to facilitate selection. You can also select other instance types.

Parameter	Description
Price Limit	The maximum hourly price for a preemptible ECS instance of a specific instance type. Before you specify the price, you can click **Price History** in the **Actions** corresponding to the instance type to view its price trend. Specify an appropriate hourly price for a preemptible instance of each instance type. Inappropriate prices may result in out-of-control costs or cause a failure to meet the target capacity. You can use one of the following methods to specify a maximum hourly price to bid for preemptible instances of an instance type: ■ Select Automatic Bidding. The real-time market price is used as the maximum hourly price for the bid. This way, preemptible ECS instances will not fail to be created due to a low bid, and the instance costs change with the market price. ■ Choose Set Maximum Price > Maximum Price, and then set a maximum hourly price. This way, preemptible ECS instances fail to be created if your bid is less than the market price, and out-of-control instance costs are prevented. ■ Choose Set Maximum Price > Pay-as-you-go Price, and then set a percentage of the pay-as-you-go price as the maximum hourly price. This way, you can have some savings over pay-as-you-go prices. For example, if you set the percentage to 50%, preemptible ECS instances fail to be created when the market price is higher than 50% of the pay-as-you-go price.

iii. **Preemptible Instance Interruption Settings:** The action to be taken on preemptible ECS instances when the auto provisioning group scales in. The following table describes the options for this parameter.

Option	Description
Release	Preemptible ECS instances are released.
Stop Instances	Preemptible ECS instances are stopped. The auto provisioning group first uses these instances when it scales out.

iv. **Provisioning Policy:** the policy used to create ECS 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 ECS instances. **Note** Preemptible ECS instances may be reclaimed due to market price changes or insufficient resources. The lower reclaim rate, the better.
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 multiple instance configurations. **Note** Preemptible ECS instances are reclaimed based on their instance types, and instance resources in the same instance family are shared. If you select Balanced Distribution Policy, we recommend that you configure instance types from different instance families to prevent instances being reclaimed at the same time and ensure high availability of the instance cluster.

7. Configure the parameters in the Advanced section. The following table describes the parameters.

Parameter	Description
Group Type	○ **One-time Delivery:** After the auto provisioning group is started, it attempts to create an instance cluster with the target capacity only once. If instances fail to be created, the auto provisioning group makes no attempts to create the instances again. ○ **Continuous Delivery and Maintain Capacity:** After the auto provisioning group is started, it continuously attempts to create an instance cluster with the target capacity. The auto provisioning group continuously compares the real-time capacity with the target capacity. If a gap exists between the real-time capacity and the target capacity, the auto provisioning group automatically scales in or out to meet the target capacity.

Parameter	Description
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 started immediately after it is created. ○ **Specify Start Time**: Specify a point in time at which to start the auto provisioning group.
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 unless you delete it. ○ **Specify End Time**: Specify a 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 all instance types. If the specified maximum hourly price specific to an instance type is different from the global maximum hourly price, the lower one of the two prices is used.
Instance Shutdown Settings	○ **Shut Down Instances Upon Group Expiration**: stops instances in the auto provisioning group when it expires. ○ **Shut Down Excessive Instances When Target Capacity Is Exceeded**: stops the instances that exceed the target capacity in the auto provisioning group.

8. Click **Create Provisioning Group**.

Result

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

5.4. View an auto provisioning group

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

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Deployment & Elasticity > Auto Provisioning**.

3. In the top navigation bar, select a region.
4. Click the ID of an auto provisioning group. The following table describes details of the auto provisioning group:

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

5.5. Modify an auto provisioning group

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

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Deployment & Elasticity > Auto Provisioning**.
3. In the top navigation bar, select a region.
4. Find the auto provisioning group that you want to modify, and click **Modify** in the **Actions** column.
5. Modify the parameters, and click **OK**. The following parameters can be modified. For more information, see Create an auto provisioning group.

Category	Parameter
Basic information	Group Name
Group Capacity	○ Target Capacity ○ Target Capacity of Preemptible Instances ○ Target Capacity of Pay-As-You-Go Instances **Note** Modifying the capacity of an auto provisioning group will affect scheduling tasks. After modification, the auto provisioning group, of the One-time Delivery type or the Continuous Delivery and Maintain Capacity type, will perform the scheduling task.

Category	Parameter
Capacity-related Settings	○ Default Billing Method of Supplemental Instances ○ Global Maximum Price for Preemptible Instances ○ Shut Down Instances Upon Group Expiration ○ Shut Down Excess Instances When Target Capacity Is Exceeded Note Modifying the capacity-related settings will affect scheduling tasks. After modification, the auto provisioning group, of the One-time Delivery type or the Continuous Delivery and Maintain Capacity type, will perform the scheduling task.

Related information

- ModifyAutoProvisioningGroup

5.6. Delete auto provisioning groups

This topic describes how to delete auto provisioning groups.

Procedure

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Deployment & Elasticity > Auto Provisioning**.
3. In the top navigation bar, select a region.
4. Find the auto provisioning group that you want to delete, and click **Delete** in the **Actions** column. You can also select multiple auto provisioning groups and click **Delete Group** to batch delete the groups.
5. Choose whether to delete the instances in the auto provisioning group. An auto provisioning group can mitigate the effects caused by reclaimed preemptible instances. If you choose not to delete the instances, you must monitor the running status of the preemptible instances to avoid service interruptions.
6. Click **OK**.

6.Terraform

6.1. Install and configure Terraform

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

Procedure

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

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

4. Create and authorize a RAM user to improve the flexibility and security of 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.
   iii. Authorize the RAM user. In this example, the `AliyunECSFullAccess` and `AliyunVPCFullAccess` permissions are granted to the *Terraform* 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"
   ```

6.2. Create an ECS instance

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

Procedure

1. Create a VPC and a VSwitch. Terraform 0.11 is used in this example.

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

   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 init` command to initialize the environment.

   iii. Run the `terraform plan` command to view resources.

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

   v. 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 the *terraform.tf* file, 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 = "22/22"
priority = 1
security_group_id = "${alicloud_security_group.default.id}"
cidr_ip = "0.0.0.0/0"
}
```

ii. Run the `terraform plan` command to view resources.

iii. After you confirm that the resources are correct, run the `terraform apply` command to create the security group and add the security group rule.

iv. 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 the *terraform.tf* file, add the following content:

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

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

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

ii. Run the `terraform plan` command to view resources.

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

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

v. 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_140405_64_40G_cloudinit_20161115.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 = "22/22"
priority = 1
security_group_id = "${alicloud_security_group.default.id}"
cidr_ip = "0.0.0.0/0"
}
```

6.3. Create multiple ECS instances

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

Procedure

1. Create a VPC and a VSwitch. Terraform 0.11 is used in this example.

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

   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 init` command to initialize the environment.

   iii. Run the `terraform plan` command to view resources.

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

   v. 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 the *terraform.tf* file, 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 = "internet"
policy = "accept"
port_range = "22/22"
priority = 1
security_group_id = "${alicloud_security_group.default.id}"
cidr_ip = "0.0.0.0/0"
}
```

ii. Run the `terraform plan` command to view resources.

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

iv. Run the `terraform show` command to view the created security group and added security group rule. You can also log on to the ECS console to view the security group and security group rule.

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

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

```
module "tf-instances" {
source = "alibaba/ecs-instance/alicloud"
vswitch_id = "${alicloud_vswitch.vsw.id}"
group_ids = ["${alicloud_security_group.default. *.id}"]
availability_zone = "cn-beijing-b"
disk_category = "cloud_ssd"
disk_name = "my_module_disk"
disk_size = "50"
number_of_disks = 7

instance_name = "my_module_instances_"
host_name = "sample"
internet_charge_type = "PayByTraffic"
number_of_instances = "3"
password="User@123"
}
```

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

ii. Run the `terraform plan` command to view resources.

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

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

v. 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 = "22/22"
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"
vswitch_id = "${alicloud_vswitch.vsw.id}"
group_ids = ["${alicloud_security_group.default. *.id}"]
availability_zone = "cn-beijing-b"
disk_category = "cloud_ssd"
disk_name = "my_module_disk"
disk_size = "50"
number_of_disks = 7

instance_name = "my_module_instances_"
host_name = "sample"
internet_charge_type = "PayByTraffic"
number_of_instances = "3"
password="User@123"
}
```

6.4. Deploy a web cluster

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

Context

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

Procedure

1. Create a VPC and a VSwitch. Terraform 0.11 is used in this example.

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

   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 init` command to initialize the environment.

   iii. Run the `terraform plan` command to view resources.

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

   v. 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 the *terraform.tf* file, 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 = "internet"
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 plan` command to view resources.

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

iv. Run the `terraform show` command to view the created security group and added security group rule. You can also log on to the ECS console to view the security group and security group rule.

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

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 plan` command to view resources.

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

iv. Run the `terraform show` command to view the created SLB instance. You can also log on to the SLB console to view the created SLB instance.

4. Create Auto Scaling resources.

   In this example, the following resources are created:

   - Scaling group: Specify 2 as the minimum number of instances and 10 as the maximum number of instances in the template, and attach the created SLB instance to the scaling group. Because scaling groups depend on SLB listener configurations, you must use the depends_on attribute to specify the deployment sequence in the template.
   - Scaling group configuration: Specify the specific configuration of the ECS instance in the template. The initialization configuration (user-data) generates a Hello World page and provides services over port 8080. To simplify operations, this example assigns a public IP address to the virtual machine and set `force_delete` to true to subsequently delete the environment.

- Scaling rule: Define 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 = "${alicloud_ess_scaling_group.scaling.id}"
image_id = "ubuntu_140405_64_40G_cloudinit_20161115.vhd"
instance_type = "ecs.n2.small"
security_group_id = "${alicloud_security_group.default.id}"
active=true
enable=true
user_data = "#! /bin/bash\necho \"Hello, World\" > index.html\nnohup busybox httpd -f -p 80
80&"
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 plan` command to view resources.

  iii. After you confirm that the resources are correct, run the `terraform apply` command to create the Auto Scaling resources.
  After the resources are created, the public IP address of the SLB instance is displayed.

  After two minutes, Auto Scaling will create an ECS instance.

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

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

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