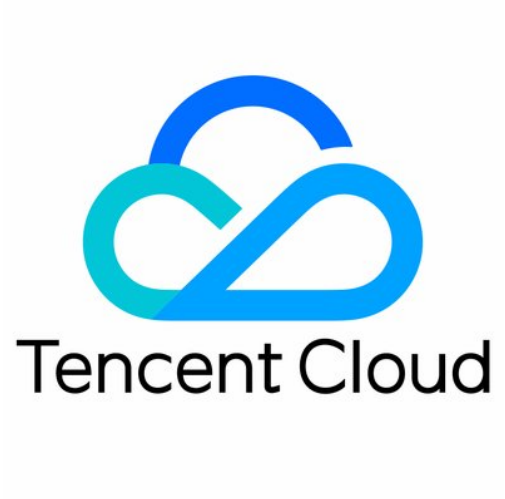
Auto Scaling

Product Introduction

Product Documentation



Contents

Product Introduction

Product Introduction

Last updated：2020-05-27 14:14:32

What is Auto Scaling (AS)?

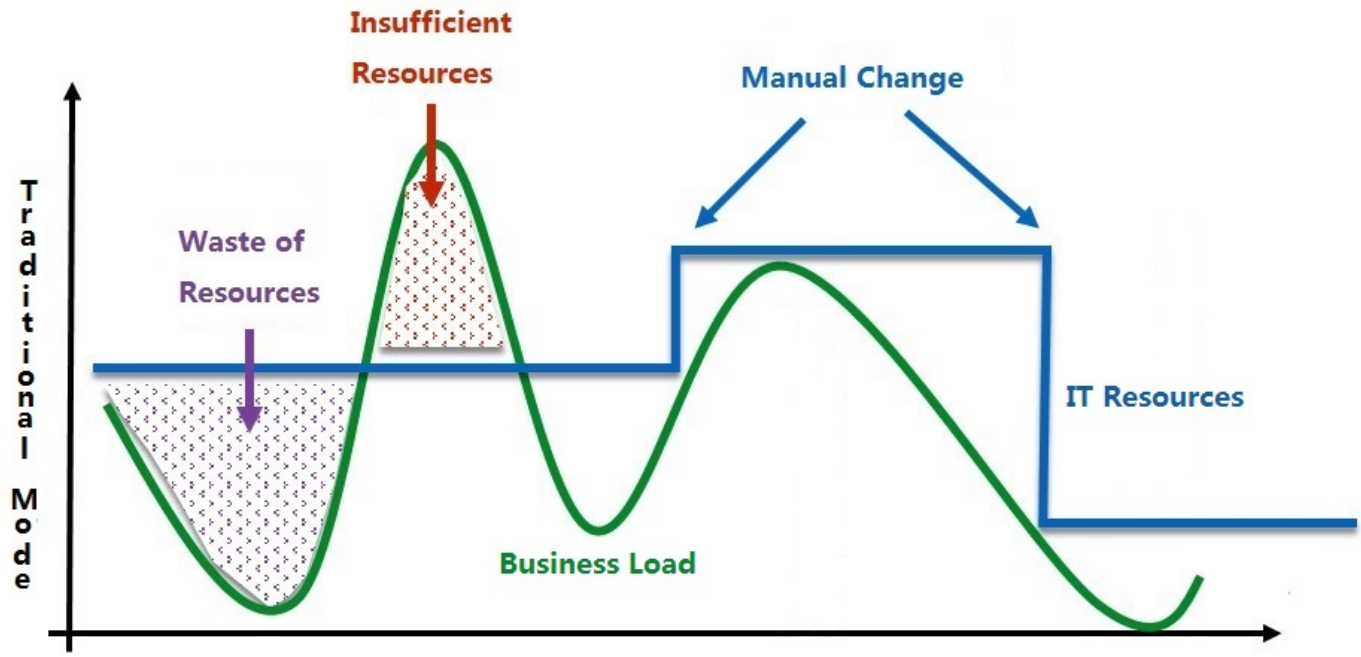
Auto Scaling (AS) can automatically adjust CVM computing resources according to configured policies and your business needs, ensuring that you always have the correct number of CVM instances available to handle the load of your applications. Intelligent scale-out and scale-in are important for cost control and resource management. When the traffic increases, you need to add more servers to process the additional load. When the traffic decreases, you need to terminate the unnecessary servers.

With AS, you only need to set up the conditions for scale-out and scale-in in advance. When the scale-out conditions are satisfied, AS will automatically add CVMs to maintain performance. When demand decreases, AS will remove CVMs according to the scale-in conditions.

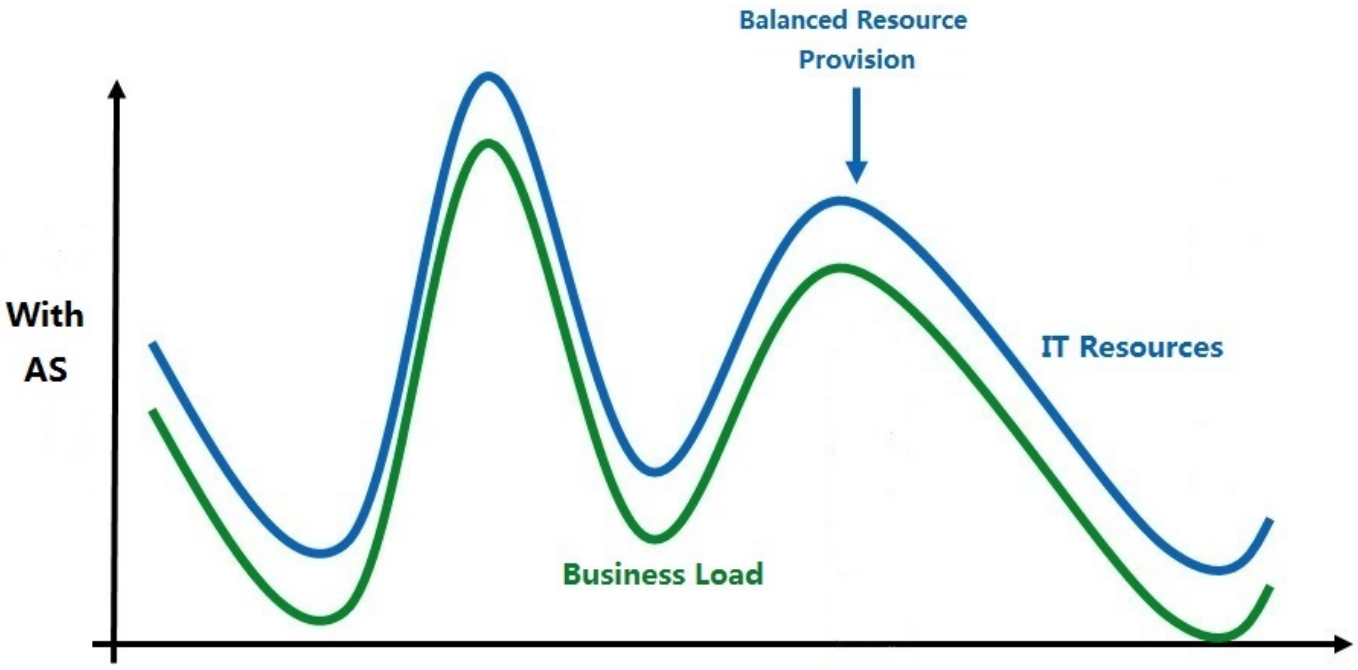
As shown in the following figures, Auto Scaling can help maintain the appropriate amount of resources, and keep your cluster stay in healthy status. You will get rid of the following troubles in the traditional model:

- Insufficient servers due to a surge in business or a CC attack, resulting in no response from your service
- Estimation of resources based on peak access volume when the access volume rarely peaks, resulting in wasted resources
- Personal surveillance and frequent handling of capacity alarms, which require multiple manual changes

Traditional mode:



With AS:



How AS Works

In common Web application services, your cluster usually runs multiple copies of an application to meet client traffic, for example, the frontend server cluster at the access layer, the application server cluster at the logical layer, and the backend cache server cluster. Every instance can process client requests.

These instances are similar or identical and the quantity is usually adjustable. You can add them to one scaling group for easy management:

- Specify the minimum number of instances in each scaling group, and AS will ensure that the number of instances in the group is never less than the minimum.
- Specify the maximum number of instances in each scaling group, and AS will ensure that the number of instances in the group is never more than the maximum.
- Specify a scaling policy, and AS will launch or terminate instances when the demands of an application increase or decrease. There are two kinds of scaling policies:
  a. Dynamic scaling policy: Scale in or out dynamically according to specified conditions (for example, scale out when the CPU utilization of the CVM instances in the scaling group is larger than 60%)
  b. Scheduled scaling policy: Scale in or out at specified times (for example, scale out every day at 21:00)
- After setting the scaling policy, you can also set the notification policies. When scaling occurs, AS informs you via e-mail, SMS and internal message. Rather than constantly focusing on your service traffic volume changes, you can just check the notifications from AS.
- You can also specify the number of machines required at any time, or add existing machines to the scaling group for joint management.

Basic concepts of Auto Scaling

Basic concepts related to Auto Scaling:

- Scaling group
- Launch configuration
- Scaling policy
- Cooldown period

1. Scaling group

A scaling group contains a collection of CVM instances that follow the same policies and have a shared purpose. Scaling groups define attributes such as the maximum and minimum numbers of CVM instances in the group and their associated CLB instances.

2. Launch configuration

A launch configuration is the template for the automatic creation of CVMs. It contains the image ID, CVM instance type, system disk/data disk types and capacities, key pair, security group, etc.

When a scaling group is created, it must be specified and cannot be edited once created.

3. Scaling policy

A scaling policy defines the conditions for executing a scaling action. The trigger condition can be a time point or an alarm of cloud monitoring, and the action can be removing or adding a CVM.
There are two types of scaling policies:

- **Scheduled scaling policy**
  CVM instances are automatically added or removed at a specified point in time, which can be performed on a periodic basis.

- **Dynamic scaling policy**
  CVM instances are automatically added or removed based on cloud monitoring metrics such as CPU, memory, and network traffic.

4. Cooldown period

Cooldown period refers to the locking duration after one scaling activity (addition or removal of CVM instances) is executed in a scaling group. The scaling group does not perform scaling activities during this duration. The cooldown period can be specified as between 0-999999 (seconds).

Product Advantages

Last updated：2020-12-08 14:29:33

Benefits	With Auto Scaling	Without Auto Scaling
Automatic	**Automatically scale instances without manual intervention** Auto Scaling (AS) automatically and dynamically adds or removes CVM instances in real time based on the business load, ensuring that you are running the optimal number of instances without manual intervention. For example, you can set a scaling policy to add CVM instances to the scaling group when the CPU utilization is high, and the new CVM instances will be charged by the second. Similarly, you can also set a policy to remove instances from the scaling group when the CPU utilization is low. If your load changes are predictable, you can set a scheduled task to plan your scaling activities. The new instances can also be directly associated with the existing cloud load balancer (CLB) to share the distributed traffic of the scaling group and to improve service availability. You can also send an alarm to Admin to keep an eye on any exceptions.	**Cumbersome manual operations are needed** The resources are subject to manual creation and termination. The CLB needs to be configured manually. Manual operation is error-prone, which may affect the business.
Cost Saving	**Appropriate scaling of instances reduces costs** AS helps you maintain an optimal number of instances in response to changing business demands. AS launches new CVM instances seamlessly and automatically when demand increases, and terminates unneeded CVM instances automatically when demand decreases. This improves device utilization and reduces the costs of deployment and instances.	**Idle resources result in waste** Extra CVMs need to be reserved to ensure the application always has enough capacity to meet demand.

Benefits	With Auto Scaling	Without Auto Scaling
Fault Tolerance	**AS automatically monitors the health of instances** AS detects any unhealthy instance, and automatically replaces it with a new one. This ensures that your application is getting the desired computing capacity so that your business can run normally and smoothly.	**Unhealthy instances are exposed to delayed processing** Unhealthy instances are not replaced until the business interruption is discovered, which compromises business availability.

Application Scenario

Last updated：2020-02-19 17:49:10

1. Configuring scaling policy in advance

If you know when to scale out and scale in, you can set an Auto Scaling scheduling policy. At the corresponding time, the system will automatically add or remove a CVM instance, with no need for human efforts.

2. Coping with business volume surges with low costs

When the traffic arises, you need to prepare CVMs in advance to prevent CVM overload caused by the sudden surge in CPU utilization. When the traffic decrease, you need to reduce CVMs accordingly. With the Auto Scaling policy configured, the system will determine when to trigger scaling accordingly. When the monitoring metrics reach the threshold, then CVM instances are automatically added or removed, and Cloud Load Balancer configuration is automatically implemented. This reduces the cost for servers and also human efforts.

3. Replacing unhealthy CVMs automatically

To prevent your business from being affected by unhealthy CVMs, you need to constantly monitor the operation status of CVMs. With Auto Scaling, the system regularly performs health checks on CVMs. When the system detects an exceptional CVM instance, it automatically creates a new instance to replace the exceptional instance. You can check all the logs if necessary.

Service Limits

Last updated：2022-07-29 14:42:51

Auto Scaling is now available in all regions except edge-server regions. The following table details the use limits of this feature:

Limit Type	Remarks
One user under one region	• Up to 50 launch configurations can be created. • Up to 50 scaling groups can be created. • The maximum number of CVM instances that can be created depends on your CVM quota. For more information, see Purchase Limits for Pay-as-You-Go CVM Instances.
A scaling group	• There can be only one launch configuration. • Up to 2,000 CVM instances can be auto-scaled. • Up to 100 scaling policies and 10 scheduled actions can be created. • Up to 5 notifications can be created. • Up to 10 lifecycle hooks can be created.
Others	• The number of CVMs in all scaling groups cannot exceed the maximum number of IP addresses that the VPC subnet can provide. • Currently, Auto Scaling does not support scaling up, which means it cannot automatically scale up the CPU, memory, or bandwidth of CVM instances. • Auto Scaling and launch configurations are services supported at the region level. Therefore, you can only launch or terminate CVM instances in the same region. • A scaling group and its associated CLB instances (in the case of a cross-region CLB instance, its backend VPC) must be in the same network environment (the VPC instance or the basic network in the same region).

Access Management

Last updated：2023-02-07 17:39:52

Overview

Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you with the security management of access permissions for resources under your Tencent Cloud account. With CAM, you can create, manage, and terminate users or user groups, and can use identity and policy management to control the permissions other users have to use Tencent Cloud resources. Policies can be used to authorize or block the use of specified resources by users to complete specified tasks. When you use CAM, you can associate policies with a user or user group to perform permissions control.

Auto Scaling is already integrated with CAM. You can use CAM to control the permissions for resources related to Auto Scaling.

Related Concepts

CAM users

CAM users are entities that you create in Tencent Cloud, and each CAM user is associated with only one Tencent Cloud account. The Tencent Cloud account you register is the **root account**. You can also create **sub-accounts** with different permissions in Cloud Access Management. There are 3 types of sub-accounts: sub-users, collaborators, and message recipients.

Policies

The policy is the syntax rule used to define and describe one or more permissions. CAM supports two types of policies: preset policy and custom policy.

- Preset policies: policies created and managed by Tencent Cloud. These are some common permission sets that are frequently used by users, such as full read and write permissions for resources. Preset policies have a wide range of operation objects, coarse operation granularity, and are preset by the system. They cannot be edited by users.
- Custom policies: policies created by users. These permit fine-grained division of permissions. For example, a usage policy is associated with a sub-account that gives the sub-account management permissions for the scaling groups of Auto Scaling, but no management permissions for TencentDB instances.

Resources

Resource is an element of policies that describes one or multiple operation objects. For example, the launch configuration and scaling groups of Auto Scaling.

Preset Policies of Auto Scaling

Preset policy	Permissions granted
QcloudASFullAccess	After association, full read-write access to Auto Scaling (AS) is obtained
QcloudASReadOnlyAccess	After association, read-only access to Auto Scaling (AS) is obtained

Authorizable Resource Types

Resource-level permissions specify which resources a user can operate. For example, you can authorize a user to have operation permissions for scaling groups in Guangzhou region.
The resource types of Auto Scaling that can be authorized through CAM are as follows:

Resource type	Resource description method in authorization policy
Launch configuration	`qcs::as:$region:$account:launch-configuration/*`
Scaling group	`qcs::as:$region:$account:auto-scaling-group/*`

The following table lists the resource-level permissions operations APIs that Auto Scaling supports, as well as the resource path supported by each operation.
When setting the resource path, you must modify variable parameters such as `$region`, `$account`、`$LaunchConfigurationId`, and `$AutoScalingGroupId` according to your actual parameter information. You can also use `*` in the path as a wildcard character.
For information about related concepts in CAM policies such as region, action, account, and resource, see Resource Description Method.

API: action	Resource path: resource
CreateLaunchConfiguration	`qcs::as:$region:$account:launch-configuration/*`
DeleteLaunchConfiguration	`qcs::as:$region:$account:launch-configuration/$LaunchConfigurationId`

API: action	Resource path: resource
DescribeLaunchConfigurations	`qcs::as:$region:$account:launch-configuration/*` `qcs::as:$region:$account:launch-configuration/$LaunchConfigurationId`
ModifyLaunchConfigurationAttributes	`qcs::as:$region:$account:launch-configuration/$LaunchConfigurationId`
UpgradeLaunchConfiguration	`qcs::as:$region:$account:launch-configuration/$LaunchConfigurationId`
CreateAutoScalingGroup	`qcs::as:$region:$account:auto-scaling-group/*`
CreateAutoScalingGroupFromInstance	`qcs::as:$region:$account:auto-scaling-group/*`
DeleteAutoScalingGroup	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DescribeAutoScalingGroups	`qcs::as:$region:$account:auto-scaling-group/*` `qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
ModifyAutoScalingGroup	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
ModifyLoadBalancers	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
EnableAutoScalingGroup	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DisableAutoScalingGroup	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
ModifyDesiredCapacity	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DescribeAutoScalingActivities	`qcs::as:$region:$account:auto-scaling-group/*` `qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
AttachInstances	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`

API: action	Resource path: resource
DetachInstances	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
RemoveInstances	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DescribeAutoScalingInstances	`qcs::as:$region:$account:auto-scaling-group/*` `qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
SetInstancesProtection	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
CreateScheduledAction	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DeleteScheduledAction	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DescribeScheduledActions	`qcs::as:$region:$account:auto-scaling-group/*` `qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
ModifyScheduledAction	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
CreateScalingPolicy	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DeleteScalingPolicy	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DescribeScalingPolicies	`qcs::as:$region:$account:auto-scaling-group/*` `qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
ModifyScalingPolicy	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
ExecuteScalingPolicy	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
CreateNotificationConfiguration	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`

API: action	Resource path: resource
DeleteNotificationConfiguration	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DescribeNotificationConfigurations	`qcs::as:$region:$account:auto-scaling-group/*` `qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
ModifyNotificationConfiguration	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
CreateLifecycleHook	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DeleteLifecycleHook	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DescribeLifecycleHooks	`qcs::as:$region:$account:auto-scaling-group/*` `qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
UpgradeLifecycleHook	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
CompleteLifecycleAction	`qcs::as:$region:$account:auto-scaling-group/$AutoScalingGroupId`
DescribeAccountLimits	`*`

Auto Scaling CAM Policy Example

The following section provides specific examples that display how to use CAM to control permissions for Auto Scaling:

- Create policy: Guangzhou region allows access permissions to all scaling groups.

```
# In this example, `$account` must be substituted with the account information
{
"version": "2.0",
"statement": [
{
"effect": "allow",
"action": [
```

```
"name/as:*"
],
"resource": [
"qcs::as:ap-guangzhou:$account:auto-scaling-group/*"
]
}
]
}
```

- Create policy: Guangzhou region prohibits access permissions to a certain scaling group.

```
# In this example, `$account` must be substituted with the account information,
and `$AutoScalingGroupId` must be substituted with the corresponding AutoScalin
gGroupId.
{
"version": "2.0",
"statement": [
{
"effect": "deny",
"action": [
"name/as:*"
],
"resource": [
"qcs::as:ap-guangzhou:$account:auto-scaling-group/$AutoScalingGroupId"
]
}
]
}
```

- Create policy: all read APIs have access permissions in all regions.

```
{
"version": "2.0",
"statement": [
{
"effect": "allow",
"action": [
"name/as:Describe*"
],
"resource": [
"*"
]
}
```

```
]
}
```