

弹性伸缩 API 文档



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

API 文档

更新历史

简介

API 概览

调用方式

请求结构

公共参数

签名方法 v3

签名方法

返回结果

参数类型

伸缩组相关接口

修改期望实例数

根据实例创建启动配置及伸缩组

启用伸缩组

停用伸缩组

修改伸缩组的负载均衡器

修改负载均衡器目标规则属性

移出负载均衡器

添加负载均衡器

查询伸缩活动

查询伸缩组最新一次伸缩活动

删除伸缩组

修改伸缩组

查询实例

查询伸缩组

创建伸缩组

指定数量扩容实例

指定数量缩容实例

查询弹性伸缩配置建议

伸缩组实例相关接口

从伸缩组中删除 CVM 实例

从伸缩组移出 CVM 实例

添加 CVM 实例到伸缩组

设置实例保护

关闭伸缩组内 CVM 实例

开启伸缩组内 CVM 实例

实例退出备用中状态

实例进入备用中状态

启动配置相关接口

查询启动配置

删除启动配置

创建启动配置

修改启动配置属性

清除启动配置属性

定时任务相关接口

修改定时任务

查询定时任务

删除定时任务

创建定时任务

告警触发策略相关接口

修改告警触发策略

查询告警触发策略

删除告警触发策略

创建告警触发策略

触发伸缩策略

通知相关接口

修改通知

查询通知

删除通知

创建通知

生命周期挂钩相关接口

升级生命周期挂钩

查询生命周期挂钩

删除生命周期挂钩

创建生命周期挂钩

完成生命周期动作

修改生命周期挂钩

实例刷新相关接口

暂停实例刷新

启动实例刷新

回滚实例刷新

恢复实例刷新

查询实例刷新活动

取消实例刷新

其他接口

查询用户账号的资源限制

数据结构

错误码

API 文档

更新历史

最近更新时间：2025-05-14 01:09:29

第 84 次发布

发布时间：2025-05-14 01:09:15

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeAutoScalingGroupLastActivities](#)
 - 新增入参：ExcludeCancelledActivity

第 83 次发布

发布时间：2025-03-17 01:08:05

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [EnterStandby](#)

第 82 次发布

发布时间：2025-03-14 01:08:03

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Instance](#)
 - 修改成员：DisasterRecoverGroupIds

第 81 次发布

发布时间：2025-01-17 01:08:27

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ClearLaunchConfigurationAttributes](#)
 - 新增入参：ClearMetadata
- [CreateLaunchConfiguration](#)
 - 新增入参：Metadata
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参：Metadata

新增数据结构：

- [Metadata](#)

- [MetadataItem](#)

第 80 次发布

发布时间：2025-01-09 01:09:15

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [RollingUpdateSettings](#)
 - 新增成员：FailProcess

第 79 次发布

发布时间：2024-12-30 01:10:05

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ServiceSettings](#)
 - 新增成员：DesiredCapacitySyncWithMaxMinSize

第 78 次发布

发布时间：2024-12-02 01:07:50

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ServiceSettings](#)
 - 新增成员：AutoUpdateInstanceTags

第 77 次发布

发布时间：2024-11-19 01:09:17

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLaunchConfiguration](#)
 - 新增入参：DedicatedClusterId
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参：DedicatedClusterId

修改数据结构：

- [LaunchConfiguration](#)
 - 新增成员：DedicatedClusterId

第 76 次发布

发布时间：2024-11-13 01:09:20

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Advice](#)
 - 修改成员：Level
- [ScalingPolicy](#)
 - 修改成员：ScalingPolicyType

第 75 次发布

发布时间：2024-11-01 01:05:15

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [RollingUpdateSettings](#)
 - 新增成员：MaxSurge

第 74 次发布

发布时间：2024-09-30 01:07:57

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLaunchConfiguration](#)
 - 新增入参：ImageFamily
 - 修改入参：ImageId
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参：ImageFamily

修改数据结构：

- [LaunchConfiguration](#)
 - 新增成员：ImageFamily

第 73 次发布

发布时间：2024-09-20 12:54:26

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ServiceSettings](#)
 - 新增成员：ReplaceMode

第 72 次发布

发布时间：2024-08-08 01:09:56

本次发布包含了以下内容：

改善已有的文档。

修改接口:

- [ClearLaunchConfigurationAttributes](#)
 - 新增入参: ClearInstanceTags
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参: InstanceTags

修改数据结构:

- [HostNameSettings](#)
 - 新增成员: HostNameSuffix

第 71 次发布

发布时间: 2024-07-23 01:04:47

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [DataDisk](#)
 - 新增成员: BurstPerformance
- [InstanceNameSettings](#)
 - 新增成员: InstanceNameSuffix

第 70 次发布

发布时间: 2024-01-31 01:06:41

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [LaunchConfiguration](#)
 - 新增成员: DisasterRecoverGroupIds

第 69 次发布

发布时间: 2023-11-10 01:06:51

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateAutoScalingGroup](#)
 - 新增入参: InstanceNameIndexSettings
- [ModifyAutoScalingGroup](#)
 - 新增入参: InstanceNameIndexSettings

新增数据结构:

- [InstanceNameIndexSettings](#)

修改数据结构:

- [AutoScalingGroup](#)
 - 新增成员: InstanceNameIndexSettings

第 68 次发布

发布时间：2023-09-28 01:06:16

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CancelInstanceRefresh](#)
- [DescribeRefreshActivities](#)
- [ExitStandby](#)
- [ResumeInstanceRefresh](#)
- [RollbackInstanceRefresh](#)
- [StartInstanceRefresh](#)
- [StopInstanceRefresh](#)

新增数据结构：

- [RefreshActivity](#)
- [RefreshBatch](#)
- [RefreshBatchRelatedInstance](#)
- [RefreshSettings](#)
- [RollingUpdateSettings](#)

第 67 次发布

发布时间：2023-09-12 01:09:06

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参：LoginSettings

第 66 次发布

发布时间：2023-08-25 01:08:59

本次发布包含了以下内容：

改善已有的文档。

预下线接口：

- [UpgradeLaunchConfiguration](#)

第 65 次发布

发布时间：2023-07-19 01:07:14

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [RelatedInstance](#)

修改数据结构：

- [Activity](#)
 - 新增成员: RelatedInstanceSet
 - 废弃成员: ActivityRelatedInstanceSet

第 64 次发布

发布时间: 2023-06-15 01:02:32

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [EnhancedService](#)
 - 废弃成员: AutomationService

第 63 次发布

发布时间: 2023-06-13 01:02:52

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [Advice](#)
 - 新增成员: Level

第 62 次发布

发布时间: 2023-03-24 01:03:57

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [ClearLaunchConfigurationAttributes](#)
 - 新增入参: ClearDisasterRecoverGroupIds
- [CreateLaunchConfiguration](#)
 - 新增入参: DisasterRecoverGroupIds
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参: DisasterRecoverGroupIds

修改数据结构:

- [Instance](#)
 - 新增成员: DisasterRecoverGroupIds

第 61 次发布

发布时间: 2023-03-23 01:04:00

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateScalingPolicy](#)
 - 新增入参: ScalingPolicyType, PredefinedMetricType, TargetValue, EstimatedInstanceWarmup, DisableScaleIn

- 修改入参: AdjustmentType, AdjustmentValue, MetricAlarm
- [ModifyScalingPolicy](#)
 - 新增入参: PredefinedMetricType, TargetValue, EstimatedInstanceWarmup, DisableScaleIn

修改数据结构:

- [Instance](#)
 - 新增成员: WarmupStatus
- [MetricAlarm](#)
 - 新增成员: PreciseThreshold
- [ScalingPolicy](#)
 - 新增成员: ScalingPolicyType, PredefinedMetricType, TargetValue, EstimatedInstanceWarmup, DisableScaleIn, MetricAlarms

第 60 次发布

发布时间: 2022-11-18 06:08:39

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateLaunchConfiguration](#)
 - 新增入参: IPv6InternetAccessible
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参: IPv6InternetAccessible
- [UpgradeLaunchConfiguration](#)
 - 新增入参: IPv6InternetAccessible

新增数据结构:

- [IPv6InternetAccessible](#)

修改数据结构:

- [LaunchConfiguration](#)
 - 新增成员: IPv6InternetAccessible

第 59 次发布

发布时间: 2022-11-16 06:08:16

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [EnhancedService](#)
 - 新增成员: AutomationToolsService

第 58 次发布

发布时间: 2022-11-03 06:08:22

本次发布包含了以下内容:

改善已有的文档。

新增数据结构:

- [RunAutomationServiceEnabled](#)

修改数据结构：

- [EnhancedService](#)
 - 新增成员：AutomationService

第 57 次发布

发布时间：2022-09-02 06:09:13

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyLifecycleHook](#)
 - 新增入参：LifecycleCommand
- [UpgradeLifecycleHook](#)
 - 新增入参：LifecycleCommand

第 56 次发布

发布时间：2022-09-01 06:08:45

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [LifecycleCommand](#)
 - 修改成员：Parameters

第 55 次发布

发布时间：2022-08-30 06:08:56

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLifecycleHook](#)
 - 新增入参：LifecycleCommand

新增数据结构：

- [LifecycleCommand](#)

修改数据结构：

- [LifecycleActionResultInfo](#)
 - 新增成员：InvocationId, InvokeCommandResult
- [LifecycleHook](#)
 - 新增成员：LifecycleCommand

第 54 次发布

发布时间：2022-08-11 06:10:05

本次发布包含了以下内容：

改善已有的文档。

修改接口:

- [CreateLaunchConfiguration](#)
 - 新增入参: HpcClusterId
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参: HpcClusterId

修改数据结构:

- [LaunchConfiguration](#)
 - 新增成员: HpcClusterId

第 53 次发布

发布时间: 2022-07-26 06:06:18

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateLaunchConfiguration](#)
 - 新增入参: Tags

新增数据结构:

- [InvocationResult](#)

修改数据结构:

- [Activity](#)
 - 新增成员: InvocationResultSet
- [LaunchConfiguration](#)
 - 新增成员: Tags

第 52 次发布

发布时间: 2022-04-28 06:06:38

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [DataDisk](#)
 - 新增成员: ThroughputPerformance

第 51 次发布

发布时间: 2022-04-25 06:01:06

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [ModifyLifecycleHook](#)

修改数据结构:

- [AutoScalingAdvice](#)

- 新增成员: Level

第 50 次发布

发布时间: 2022-04-04 06:07:52

本次发布包含了以下内容:

改善已有的文档。

删除接口:

- DescribePaiInstances

删除数据结构:

- PaiInstance

第 49 次发布

发布时间: 2022-03-30 06:01:19

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [DataDisk](#)
 - 新增成员: Encrypt

第 48 次发布

发布时间: 2022-03-11 08:03:26

本次发布包含了以下内容:

改善已有的文档。

删除接口:

- PreviewPaiDomainName

第 47 次发布

发布时间: 2022-03-09 08:05:55

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [DataDisk](#)
 - 新增成员: DeleteWithInstance
- [ScheduledAction](#)
 - 新增成员: ScheduledType

第 46 次发布

发布时间: 2022-02-23 08:02:26

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [ModifyDesiredCapacity](#)
 - 新增入参: MinSize, MaxSize

第 45 次发布

发布时间: 2022-01-07 08:03:22

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参: CamRoleName

第 44 次发布

发布时间: 2022-01-05 08:03:39

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [AttachLoadBalancers](#)
- [DetachLoadBalancers](#)
- [ModifyLoadBalancerTargetAttributes](#)

新增数据结构:

- [ForwardLoadBalancerIdentification](#)

第 43 次发布

发布时间: 2021-12-14 08:05:13

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [DescribeAutoScalingAdvices](#)

新增数据结构:

- [Advice](#)
- [AutoScalingAdvice](#)

第 42 次发布

发布时间: 2021-11-15 08:04:22

本次发布包含了以下内容:

改善已有的文档。

删除接口:

- [CreatePailInstance](#)

第 41 次发布

发布时间：2021-10-27 08:02:46

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参：EnhancedService

第 40 次发布

发布时间：2021-10-21 11:15:03

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateNotificationConfiguration](#)
 - 新增入参：TargetType, QueueName, TopicName
 - 修改入参：NotificationUserGroupIds
- [ModifyNotificationConfiguration](#)
 - 新增入参：QueueName, TopicName

修改数据结构：

- [AutoScalingNotification](#)
 - 新增成员：TargetType, QueueName, TopicName

第 39 次发布

发布时间：2021-09-06 08:03:06

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [DetailedStatusMessage](#)

修改数据结构：

- [Activity](#)
 - 新增成员：DetailedStatusMessageSet

第 38 次发布

发布时间：2021-08-06 08:03:03

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateAutoScalingGroup](#)
 - 新增入参：CapacityRebalance
- [ModifyAutoScalingGroup](#)
 - 新增入参：CapacityRebalance

第 37 次发布

发布时间：2021-07-20 08:04:50

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateAutoScalingGroup](#)
 - 新增入参：InstanceAllocationPolicy, SpotMixedAllocationPolicy
- [ModifyAutoScalingGroup](#)
 - 新增入参：InstanceAllocationPolicy, SpotMixedAllocationPolicy

新增数据结构：

- [SpotMixedAllocationPolicy](#)

修改数据结构：

- [AutoScalingGroup](#)
 - 新增成员：InstanceAllocationPolicy, SpotMixedAllocationPolicy, CapacityRebalance

第 36 次发布

发布时间：2021-07-09 08:02:59

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ClearLaunchConfigurationAttributes](#)
 - 新增入参：ClearHostNameSettings, ClearInstanceNameSettings
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参：HostNameSettings, InstanceNameSettings

修改数据结构：

- [AutoScalingGroup](#)
 - 新增成员：HealthCheckType, LoadBalancerHealthCheckGracePeriod

第 35 次发布

发布时间：2021-06-17 08:02:54

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateAutoScalingGroup](#)
 - 新增入参：HealthCheckType, LoadBalancerHealthCheckGracePeriod
- [ModifyAutoScalingGroup](#)
 - 新增入参：HealthCheckType, LoadBalancerHealthCheckGracePeriod

修改数据结构：

- [ServiceSettings](#)
 - 新增成员：ReplaceLoadBalancerUnhealthy

第 34 次发布

发布时间：2021-05-28 08:02:36

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ScaleInInstances](#)
- [ScaleOutInstances](#)

第 33 次发布

发布时间：2021-05-11 08:02:28

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [LaunchConfiguration](#)
 - 新增成员：DiskTypePolicy

第 32 次发布

发布时间：2021-04-01 08:02:04

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ClearLaunchConfigurationAttributes](#)

修改接口：

- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参：SystemDisk, DataDisks

第 31 次发布

发布时间：2021-03-22 08:02:16

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLaunchConfiguration](#)
 - 新增入参：DiskTypePolicy
- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参：DiskTypePolicy
- [UpgradeLaunchConfiguration](#)
 - 新增入参：DiskTypePolicy

第 30 次发布

发布时间：2021-01-14 08:01:59

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参: SecurityGroupIds, InternetAccessible, InstanceChargeType, InstanceChargePrepaid, InstanceMarketOptions

第 29 次发布

发布时间: 2020-06-10 08:03:37

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ExecuteScalingPolicy](#)
 - 新增入参: TriggerSource

第 28 次发布

发布时间: 2020-06-09 08:04:26

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [InternetAccessible](#)
 - 新增成员: BandwidthPackageId

第 27 次发布

发布时间: 2020-04-20 08:03:10

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [LifecycleActionResultInfo](#)

修改数据结构：

- [Activity](#)
 - 新增成员: LifecycleActionResultSet

第 26 次发布

发布时间: 2020-03-06 14:06:29

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLifecycleHook](#)
 - 新增入参: LifecycleTransitionType
- [UpgradeLifecycleHook](#)
 - 新增入参: LifecycleTransitionType

第 25 次发布

发布时间：2020-02-26 21:50:31

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [LifecycleHook](#)
 - 新增成员：LifecycleTransitionType

第 24 次发布

发布时间：2020-02-20 10:07:02

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpgradeLaunchConfiguration](#)
 - 修改入参：InstanceNameSettings

修改数据结构：

- [ForwardLoadBalancer](#)
 - 新增成员：Region
- [LaunchConfiguration](#)
 - 修改成员：InstanceNameSettings

第 23 次发布

发布时间：2020-01-09 11:28:08

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateAutoScalingGroup](#)
 - 新增入参：MultiZoneSubnetPolicy
- [CreateLaunchConfiguration](#)
 - 新增入参：InstanceNameSettings, InstanceChargePrepaid
- [ModifyAutoScalingGroup](#)
 - 新增入参：MultiZoneSubnetPolicy
- [UpgradeLaunchConfiguration](#)
 - 新增入参：InstanceNameSettings, InstanceChargePrepaid

新增数据结构：

- [InstanceNameSettings](#)

修改数据结构：

- [AutoScalingGroup](#)
 - 新增成员：MultiZoneSubnetPolicy
- [LaunchConfiguration](#)
 - 新增成员：InstanceNameSettings, InstanceChargePrepaid

第 22 次发布

发布时间：2019-12-17 12:00:22

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [StartAutoScalingInstances](#)
- [StopAutoScalingInstances](#)

修改接口：

- [CreateAutoScalingGroup](#)
 - 新增入参：Ipv6AddressCount
- [ModifyAutoScalingGroup](#)
 - 新增入参：Ipv6AddressCount

修改数据结构：

- [AutoScalingGroup](#)
 - 新增成员：Ipv6AddressCount
- [ServiceSettings](#)
 - 新增成员：ScalingMode

第 21 次发布

发布时间：2019-11-28 22:15:47

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateAutoScalingGroupFromInstance](#)

修改数据结构：

- [PaiInstance](#)
 - 新增成员：PaiMateUrl

第 20 次发布

发布时间：2019-11-21 19:39:04

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLaunchConfiguration](#)
 - 新增入参：HostNameSettings
- [UpgradeLaunchConfiguration](#)
 - 新增入参：HostNameSettings

新增数据结构：

- [HostNameSettings](#)

修改数据结构：

- [LaunchConfiguration](#)
 - 新增成员: HostNameSettings

第 19 次发布

发布时间: 2019-09-26 21:43:04

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [ModifyLoadBalancers](#)
 - 新增入参: LoadBalancersCheckPolicy

修改数据结构:

- [LaunchConfiguration](#)
 - 新增成员: LastOperationInstanceTypesCheckPolicy

第 18 次发布

发布时间: 2019-08-29 20:13:44

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [DescribeAutoScalingGroupLastActivities](#)

第 17 次发布

发布时间: 2019-08-01 14:33:42

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [Instance](#)
 - 新增成员: AutoScalingGroupName

第 16 次发布

发布时间: 2019-07-12 11:37:21

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateAutoScalingGroup](#)
 - 新增入参: ServiceSettings
- [CreateLaunchConfiguration](#)
 - 新增入参: CamRoleName
- [ModifyAutoScalingGroup](#)
 - 新增入参: ServiceSettings
- [UpgradeLaunchConfiguration](#)
 - 新增入参: CamRoleName

新增数据结构：

- [ServiceSettings](#)

修改数据结构：

- [AutoScalingGroup](#)
 - 新增成员：ServiceSettings
- [LaunchConfiguration](#)
 - 新增成员：CamRoleName

第 15 次发布

发布时间：2019-06-13 16:13:04

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ExecuteScalingPolicy](#)

第 14 次发布

发布时间：2019-06-06 20:15:07

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateAutoScalingGroup](#)
 - 新增入参：Tags

新增数据结构：

- [Tag](#)

修改数据结构：

- [AutoScalingGroup](#)
 - 新增成员：Tags

第 13 次发布

发布时间：2019-05-30 19:51:36

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [UpgradeLaunchConfiguration](#)

修改接口：

- [AttachInstances](#)
 - 新增出参：ActivityId
- [DetachInstances](#)
 - 新增出参：ActivityId
- [RemoveInstances](#)

- 新增出参: ActivityId

修改数据结构:

- [LifecycleHook](#)
 - 修改成员: NotificationTarget
- [ScalingPolicy](#)
 - 修改成员: AdjustmentValue

第 12 次发布

发布时间: 2019-05-16 19:38:05

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [Instance](#)
 - 新增成员: VersionNumber
- [LaunchConfiguration](#)
 - 新增成员: VersionNumber, UpdatedTime

第 11 次发布

发布时间: 2019-04-25 19:24:45

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [CompleteLifecycleAction](#)
- [CreateLifecycleHook](#)
- [CreatePailInstance](#)
- [DeleteLifecycleHook](#)
- [DescribeLifecycleHooks](#)
- [DescribePailInstances](#)
- [PreviewPaiDomainName](#)
- [UpgradeLifecycleHook](#)

修改接口:

- [CreateLaunchConfiguration](#)
 - 新增入参: InstanceTags

新增数据结构:

- [InstanceChargePrepaid](#)
- [InstanceTag](#)
- [LifecycleHook](#)
- [NotificationTarget](#)
- [PailInstance](#)

修改数据结构:

- [LaunchConfiguration](#)
 - 新增成员: InstanceTags

第 10 次发布

发布时间：2019-03-08 17:33:58

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [AutoScalingGroup](#)
 - 新增成员：InActivityStatus

第 9 次发布

发布时间：2019-02-22 14:42:40

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateNotificationConfiguration](#)
- [CreateScalingPolicy](#)
- [DeleteNotificationConfiguration](#)
- [DeleteScalingPolicy](#)
- [DescribeNotificationConfigurations](#)
- [DescribeScalingPolicies](#)
- [ModifyNotificationConfiguration](#)
- [ModifyScalingPolicy](#)
- [SetInstancesProtection](#)

修改接口：

- [DescribeAutoScalingActivities](#)
 - 新增入参：StartTime, EndTime

新增数据结构：

- [AutoScalingNotification](#)
- [MetricAlarm](#)
- [ScalingPolicy](#)

第 8 次发布

发布时间：2019-02-14 16:01:19

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [AutoScalingGroup](#)
 - 修改成员：DefaultCooldown, DesiredCapacity, InstanceCount, InServiceInstanceCount, MaxSize, MinSize, ProjectId

第 7 次发布

发布时间：2019-01-24 17:23:35

本次发布包含了以下内容：

改善已有的文档。

新增接口:

- [ModifyLoadBalancers](#)

修改接口:

- [ModifyLaunchConfigurationAttributes](#)
 - 新增入参: UserData

第 6 次发布

发布时间: 2018-12-27 19:18:13

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [DataDisk](#)
 - 新增成员: SnapshotId

第 5 次发布

发布时间: 2018-12-19 10:09:33

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [DescribeAutoScalingActivities](#)

新增数据结构:

- [Activity](#)
- [ActivityRelatedInstance](#)

第 4 次发布

发布时间: 2018-12-06 18:57:45

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [ModifyLaunchConfigurationAttributes](#)

修改接口:

- [CreateAutoScalingGroup](#)
 - 新增入参: ZonesCheckPolicy
- [CreateLaunchConfiguration](#)
 - 新增入参: InstanceTypesCheckPolicy
- [ModifyAutoScalingGroup](#)
 - 新增入参: ZonesCheckPolicy

第 3 次发布

发布时间: 2018-11-15 15:42:48

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateAutoScalingGroup](#)
 - 新增入参：RetryPolicy
- [CreateLaunchConfiguration](#)
 - 新增入参：InstanceTypes
 - 修改入参：InstanceType
- [ModifyAutoScalingGroup](#)
 - 新增入参：RetryPolicy

修改数据结构：

- [AutoScalingGroup](#)
 - 新增成员：RetryPolicy
- [Instance](#)
 - 新增成员：InstanceType
- [LaunchConfiguration](#)
 - 新增成员：InstanceTypes

第 2 次发布

发布时间：2018-10-19 17:44:13

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLaunchConfiguration](#)
 - 新增入参：InstanceChargeType, InstanceMarketOptions

新增数据结构：

- [InstanceMarketOptionsRequest](#)
- [SpotMarketOptions](#)

修改数据结构：

- [Instance](#)
 - 修改成员：CreationType
- [LaunchConfiguration](#)
 - 新增成员：InstanceChargeType, InstanceMarketOptions

第 1 次发布

发布时间：2018-10-11 15:37:12

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AttachInstances](#)
- [CreateAutoScalingGroup](#)
- [CreateLaunchConfiguration](#)
- [CreateScheduledAction](#)

- [DeleteAutoScalingGroup](#)
- [DeleteLaunchConfiguration](#)
- [DeleteScheduledAction](#)
- [DescribeAccountLimits](#)
- [DescribeAutoScalingGroups](#)
- [DescribeAutoScalingInstances](#)
- [DescribeLaunchConfigurations](#)
- [DescribeScheduledActions](#)
- [DetachInstances](#)
- [DisableAutoScalingGroup](#)
- [EnableAutoScalingGroup](#)
- [ModifyAutoScalingGroup](#)
- [ModifyDesiredCapacity](#)
- [ModifyScheduledAction](#)
- [RemoveInstances](#)

新增数据结构：

- [AutoScalingGroup](#)
- [AutoScalingGroupAbstract](#)
- [DataDisk](#)
- [EnhancedService](#)
- [Filter](#)
- [ForwardLoadBalancer](#)
- [Instance](#)
- [InternetAccessible](#)
- [LaunchConfiguration](#)
- [LimitedLoginSettings](#)
- [LoginSettings](#)
- [RunMonitorServiceEnabled](#)
- [RunSecurityServiceEnabled](#)
- [ScheduledAction](#)
- [SystemDisk](#)
- [TargetAttribute](#)

简介

最近更新时间：2025-04-25 01:08:54

欢迎使用腾讯云弹性伸缩（AS）！

弹性伸缩（AS）可以根据用户的业务需求和策略，自动调整计算资源。弹性伸缩根据用户设置的定时策略、周期策略或监控策略，恰到好处地增加或减少 CVM 实例并完成配置，如在业务需求高峰增长时自动增加 CVM 实例，保证业务平稳健康运行。在业务需求下降时自动减少 CVM 实例，节约资源和成本。

注意：

- 本章节弹性伸缩 API 接口均为最新 API 3.0 接口，后续弹性伸缩相关新增功能都会在此章节更新。我们强烈推荐您使用最新 API 3.0 接口。

API 概览

最近更新时间：2025-03-17 01:08:16

伸缩组相关接口

接口名称	接口功能	频率限制（次/秒）
CreateAutoScalingGroupFromInstance	根据实例创建启动配置及伸缩组	20
ModifyDesiredCapacity	修改期望实例数	20
AttachLoadBalancers	添加负载均衡器	20
DetachLoadBalancers	移出负载均衡器	20
DisableAutoScalingGroup	停用伸缩组	20
EnableAutoScalingGroup	启用伸缩组	20
ModifyLoadBalancerTargetAttributes	修改负载均衡器目标规则属性	20
ModifyLoadBalancers	修改伸缩组的负载均衡器	20
DescribeAutoScalingActivities	查询伸缩活动	20
DescribeAutoScalingGroupLastActivities	查询伸缩组最新一次伸缩活动	20
DeleteAutoScalingGroup	删除伸缩组	20
CreateAutoScalingGroup	创建伸缩组	20
DescribeAutoScalingAdvices	查询弹性伸缩配置建议	20
DescribeAutoScalingGroups	查询伸缩组	60
DescribeAutoScalingInstances	查询实例	60
ModifyAutoScalingGroup	修改伸缩组	20
ScaleInInstances	指定数量缩容实例	20
ScaleOutInstances	指定数量扩容实例	20

伸缩组实例相关接口

接口名称	接口功能	频率限制（次/秒）
AttachInstances	添加 CVM 实例到伸缩组	20
DetachInstances	从伸缩组移出 CVM 实例	20
EnterStandby	实例进入备用中状态	20
ExitStandby	实例退出备用中状态	20
RemoveInstances	从伸缩组中删除 CVM 实例	20
SetInstancesProtection	设置实例保护	20
StartAutoScalingInstances	开启伸缩组内 CVM 实例	20
StopAutoScalingInstances	关闭伸缩组内 CVM 实例	20

启动配置相关接口

接口名称	接口功能	频率限制（次/秒）
ClearLaunchConfigurationAttributes	清除启动配置属性	20
CreateLaunchConfiguration	创建启动配置	20
DeleteLaunchConfiguration	删除启动配置	10
DescribeLaunchConfigurations	查询启动配置	20
ModifyLaunchConfigurationAttributes	修改启动配置属性	20
UpgradeLaunchConfiguration	升级启动配置	20

通知相关接口

接口名称	接口功能	频率限制（次/秒）
CreateNotificationConfiguration	创建通知	20
DeleteNotificationConfiguration	删除通知	20
DescribeNotificationConfigurations	查询通知	20
ModifyNotificationConfiguration	修改通知	20

告警触发策略相关接口

接口名称	接口功能	频率限制（次/秒）
CreateScalingPolicy	创建告警触发策略	20
DeleteScalingPolicy	删除告警触发策略	20
DescribeScalingPolicies	查询告警触发策略	20
ExecuteScalingPolicy	触发伸缩策略	20
ModifyScalingPolicy	修改告警触发策略	20

其他接口

接口名称	接口功能	频率限制（次/秒）
DescribeAccountLimits	查询用户账号的资源限制	20

生命周期挂钩相关接口

接口名称	接口功能	频率限制（次/秒）
CompleteLifecycleAction	完成生命周期动作	20
CreateLifecycleHook	创建生命周期挂钩	20
DeleteLifecycleHook	删除生命周期挂钩	20
DescribeLifecycleHooks	查询生命周期挂钩	20
ModifyLifecycleHook	修改生命周期挂钩	20
UpgradeLifecycleHook	升级生命周期挂钩	20

定时任务相关接口

接口名称	接口功能	频率限制（次/秒）
CreateScheduledAction	创建定时任务	20
DeleteScheduledAction	删除定时任务	20
DescribeScheduledActions	查询定时任务	20
ModifyScheduledAction	修改定时任务	20

实例刷新相关接口

接口名称	接口功能	频率限制（次/秒）
CancelInstanceRefresh	取消实例刷新	20
DescribeRefreshActivities	查询实例刷新活动	20
ResumeInstanceRefresh	恢复实例刷新	20
RollbackInstanceRefresh	回滚实例刷新	20
StartInstanceRefresh	启动实例刷新	20
StopInstanceRefresh	暂停实例刷新	20

注意：

以上给出的接口频率限制维度为 API + 接入地域 + 子账号，有关限频更多说明参考：[API 频率限制说明](#)

调用方式

请求结构

最近更新时间：2025-05-21 01:08:27

1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 `as.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `as.ap-guangzhou.tencentcloudapi.com`。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到最近的某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 `as.ap-guangzhou.tencentcloudapi.com` 是一致的。

注意：对时延敏感的业务，建议指定带地域的域名。

注意：域名是 API 的接入点，并不代表产品或者接口实际提供服务的地域。产品支持的地域列表请在调用方式/公共参数文档中查阅，接口支持的地域请在接口文档输入参数中查阅。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	<code>as.tencentcloudapi.com</code>
华南地区（广州）	<code>as.ap-guangzhou.tencentcloudapi.com</code>
华东地区（上海）	<code>as.ap-shanghai.tencentcloudapi.com</code>
华东地区（南京）	<code>as.ap-nanjing.tencentcloudapi.com</code>
华北地区（北京）	<code>as.ap-beijing.tencentcloudapi.com</code>
西南地区（成都）	<code>as.ap-chengdu.tencentcloudapi.com</code>
西南地区（重庆）	<code>as.ap-chongqing.tencentcloudapi.com</code>
港澳台地区（中国香港）	<code>as.ap-hongkong.tencentcloudapi.com</code>
亚太东南（新加坡）	<code>as.ap-singapore.tencentcloudapi.com</code>
亚太东南（雅加达）	<code>as.ap-jakarta.tencentcloudapi.com</code>
亚太东南（曼谷）	<code>as.ap-bangkok.tencentcloudapi.com</code>
亚太东北（首尔）	<code>as.ap-seoul.tencentcloudapi.com</code>
亚太东北（东京）	<code>as.ap-tokyo.tencentcloudapi.com</code>
美国东部（弗吉尼亚）	<code>as.na-ashburn.tencentcloudapi.com</code>
美国西部（硅谷）	<code>as.na-siliconvalley.tencentcloudapi.com</code>
南美地区（圣保罗）	<code>as.sa-saopaulo.tencentcloudapi.com</code>
欧洲地区（法兰克福）	<code>as.eu-frankfurt.tencentcloudapi.com</code>

注意：由于金融区和非金融区是隔离不互通的，因此当访问金融区服务时（公共参数 `Region` 为金融区地域），需要同时指定带金融区地域的域名，最好和 `Region` 的地域保持一致。

金融区接入地域	金融区域名
华东地区（上海金融）	<code>as.ap-shanghai-fsi.tencentcloudapi.com</code>
华南地区（深圳金融）	<code>as.ap-shenzhen-fsi.tencentcloudapi.com</code>

2. 通信协议

腾讯云 API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法：

- POST (推荐)
- GET

POST 请求支持的 Content-Type 类型：

- application/json (推荐)，必须使用签名方法 v3 (TC3-HMAC-SHA256)。
- application/x-www-form-urlencoded，必须使用签名方法 v1 (HmacSHA1 或 HmacSHA256)。
- multipart/form-data (仅部分接口支持)，必须使用签名方法 v3 (TC3-HMAC-SHA256)。

GET 请求的请求包大小不得超过32KB。POST 请求使用签名方法 v1 (HmacSHA1、HmacSHA256) 时不得超过1MB。POST 请求使用签名方法 v3 (TC3-HMAC-SHA256) 时支持10MB。

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2025-03-28 01:09:37

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

签名方法 v3（有时也称作 TC3-HMAC-SHA256）相比签名方法 v1（有些文档可能会简称签名方法），更安全，支持更大的请求包，支持 POST JSON 格式，性能有一定提升，推荐使用该签名方法计算签名。完整介绍详见 [签名方法 v3](#)。

注意：出于简化的目的，部分接口文档中的示例使用的是签名方法 v1 GET 请求，而不是更安全的签名方法 v3。

使用签名方法 v3 时，公共参数需要统一放到 HTTP Header 请求头部中，如下表所示：

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKID***/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKID*** 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为具体产品名，通常为域名前缀。例如，域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 as；tc3_request 为固定字符串； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要，计算过程详见 文档 。
Token	String	否	HTTP 请求头：X-TC-Token。即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	HTTP 请求头：X-TC-Language。指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表中的前十个，接口参数设置为偏移量 Offset=0，返回数量 Limit=10，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKID***/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
```

```
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

HTTP POST (application/json) 请求结构示例:

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID*/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

{"Offset":0,"Limit":10}
```

HTTP POST (multipart/form-data) 请求结构示例 (仅特定的接口支持) :

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID*/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

使用签名方法 v1 的公共参数

使用签名方法 v1 (有时会称作 HmacSHA256 和 HmacSHA1), 公共参数需要统一放到请求串中, 完整介绍详见[文档](#)

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口, 取值为 DescribeInstances。

参数名称	类型	必选	描述
Region	String	-	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见 文档 。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表，其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****

Host: cvm.tencentcloudapi.com
```

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/

Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded

Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****
*****
```

地域列表

本产品所有接口 Region 字段的可选值如下表所示。如果接口不支持该表中的所有地域，则会在接口文档中单独说明。

地域	取值
亚太东南（曼谷）	ap-bangkok
华北地区（北京）	ap-beijing

地域	取值
西南地区（成都）	ap-chengdu
西南地区（重庆）	ap-chongqing
华南地区（广州）	ap-guangzhou
港澳台地区（中国香港）	ap-hongkong
亚太东南（雅加达）	ap-jakarta
华东地区（南京）	ap-nanjing
亚太东北（首尔）	ap-seoul
华东地区（上海）	ap-shanghai
华东地区（上海金融）	ap-shanghai-fsi
华南地区（深圳金融）	ap-shenzhen-fsi
亚太东南（新加坡）	ap-singapore
亚太东北（东京）	ap-tokyo
欧洲地区（法兰克福）	eu-frankfurt
美国东部（弗吉尼亚）	na-ashburn
美国西部（硅谷）	na-siliconvalley
南美地区（圣保罗）	sa-saopaulo

签名方法 v3

最近更新时间：2024-12-25 01:10:36

以下文档说明了签名方法 v3 的签名过程，但仅在您编写自己的代码来调用腾讯云 API 时才有用。我们推荐您使用 [腾讯云 API Explorer](#)，[腾讯云 SDK](#) 和 [腾讯云命令行工具 \(TCCLI\)](#) 等开发者工具，从而无需学习如何对 API 请求进行签名。

推荐使用 API Explorer

<> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

腾讯云 API 会对每个请求进行身份验证，用户需要使用安全凭证，经过特定的步骤对请求进行签名（Signature），每个请求都需要在公共参数中指定该签名结果并以指定的方式和格式发送请求。

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份

签名确保请求是由持有有效访问密钥的人发送的。请参阅控制台 [云 API 密钥](#) 页面获取密钥相关信息。

2. 保护传输中的数据

为了防止请求在传输过程中被篡改，腾讯云 API 会使用请求参数来计算请求的哈希值，并将生成的哈希值加密后作为请求的一部分，发送到腾讯云 API 服务器。服务器会使用收到的请求参数以同样的过程计算哈希值，并验证请求中的哈希值。如果请求被篡改，将导致哈希值不一致，腾讯云 API 将拒绝本次请求。

签名方法 v3（TC3-HMAC-SHA256）功能上覆盖了以前的签名方法 v1，而且更安全，支持更大的请求，支持 JSON 格式，POST 请求支持传空数组和空字符串，性能有一定提升，推荐使用该签名方法计算签名。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以对生成签名过程进行验证，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

申请安全凭证

本文使用的安全凭证为密钥，密钥包括 SecretId 和 SecretKey。每个用户最多可以拥有两对密钥。

- SecretId：用于标识 API 调用者身份，可以简单类比为用户名。
- SecretKey：用于验证 API 调用者的身份，可以简单类比为密码。
- 用户必须严格保管安全凭证，避免泄露，否则将危及财产安全。如已泄露，请立刻禁用该安全凭证。

申请安全凭证的具体步骤如下：

1. 登录 [腾讯云管理中心控制台](#)。
2. 前往 [云API密钥](#) 的控制台页面。
3. 在 [云API密钥](#) 页面，单击【新建密钥】创建一对密钥。

签名版本 v3 签名过程

云 API 支持 GET 和 POST 请求。对于 GET 方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于 POST 方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式绝大多数接口均支持，multipart 格式只有特定接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。推荐使用 POST 请求，因为两者的结果并无差异，但 GET 请求只支持 32 KB 以内的请求包。

下面以云服务器查询广州实例列表作为例子，分步骤介绍签名的计算过程。我们选择该接口是因为：

1. 云服务器默认已开通，该接口很常用；
2. 该接口是只读的，不会改变现有资源的状态；
3. 接口覆盖的参数种类较全，可以演示包含数据结构的数组如何使用。

在示例中，不论公共参数或者接口的参数，我们尽量选择容易犯错的情况。在实际调用接口时，请根据实际情况来，每个接口的参数并不相同，不要照抄这个例子的参数和值。此外，这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数（在 HTTP 头部设置，添加 X-TC- 前缀）。

假设用户的 SecretId 和 SecretKey 分别是：AKID***** 和 *****。用户想查看广州区云服务器名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
CanonicalURI + '\n' +
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST）。此示例取值为 POST。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠 (/)。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入其他头部参与签名以提高自身请求的唯一性和安全性，此示例额外增加了接口名头部。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接； 2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。 此示例计算结果是 content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\nx-tc-action:describeinstances\n。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。 拼接规则：

字段名称	解释
	1. 头部 key 统一转成小写; 2. 多个头部 key (小写) 按照 ASCII 升序进行拼接, 并且以分号 (;) 分隔。 此示例为 content-type;host;x-tc-action
HashedRequestPayload	请求正文 (payload, 即 body, 此示例为 {"Limit": 1, "Filters": [{"Values": [{"\u672a\u547d\u540d"}, {"Name": "instance-name"}]}) 的哈希值, 计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload))), 即对 HTTP 请求正文做 SHA256 哈希, 然后十六进制编码, 最后编码串转换成小写字母。对于 GET 请求, RequestPayload 固定为空字符串。此示例计算结果是 35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064。

根据以上规则, 示例中得到的规范请求串如下:

```
POST
/

content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com
x-tc-action:describeinstances

content-type;host;x-tc-action
35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064
```

2. 拼接待签名字符串

按如下格式拼接待签名字符串:

```
StringToSign =
Algorithm + "\n" +
RequestTimestamp + "\n" +
CredentialScope + "\n" +
HashedCanonicalRequest
```

字段名称	解释
Algorithm	签名算法, 目前固定为 TC3-HMAC-SHA256。
RequestTimestamp	请求时间戳, 即请求头部的公共参数 X-TC-Timestamp 取值, 取当前时间 UNIX 时间戳, 精确到秒。此示例取值为 1551113065。
CredentialScope	凭证范围, 格式为 Date/service/tc3_request, 包含日期、所请求的服务和终止字符串 (tc3_request)。Date 为 UTC 标准时间的日期, 取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致; service 为产品名, 必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值, 计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84。

注意:

```
1. Date 必须从时间戳 X-TC-Timestamp 计算得到, 且时区为 UTC+0。如果加入系统本地时区信息, 例如东八区, 将导致白天和晚上调用成功, 但是凌晨时调用必定失败。假设时间戳为 1551113065, 在东八区的时间是 2019-02-26 00:44:25, 但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25, 而不是 2019-02-26。
```

2. **Timestamp** 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能运行一段时间后，请求失败，返回签名过期错误。

根据以上规则，示例中得到的待签名字符串如下：

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84
```

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "*****"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

派生出的密钥 `SecretDate`、`SecretService` 和 `SecretSigning` 是二进制的数，可能包含不可打印字符，将其转为十六进制字符串打印的输出分别为：
 da98fb70dcf6b112dc21038d1eeeb3a95c74b4dcb12c1131f864f6066bd02be0,
 8d70cbefb03939f929db64d32dc2ba89b1095620119fe3e050e2b18c5bd2752f,
 b596b923aad85185e2d1f6659d2a062e0a86731226e021e61bfe06f7ed05f5af。

请注意，不同的编程语言，HMAC 库函数中参数顺序可能不一样，请以实际情况为准。此处的伪代码密钥参数 `key` 在前，消息参数 `data` 在后。通常标准库函数会提供二进制格式的返回值，也可能会提供打印友好的十六进制格式的返回值，此处使用的是二进制格式。

字段名称	解释
SecretKey	原始的 SecretKey，即 *****。
Date	即 Credential 中的 Date 字段信息。此示例取值为 2019-02-25。
Service	即 Credential 中的 Service 字段信息。此示例取值为 cvm。

2) 计算签名，伪代码如下：

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
Algorithm + ' ' +
'Credential=' + SecretId + '/' + CredentialScope + ', ' +
'SignedHeaders=' + SignedHeaders + ', ' +
'Signature=' + Signature
```

字段名称	解释
Algorithm	签名方法，固定为 TC3-HMAC-SHA256。
SecretId	密钥对中的 SecretId，即 AKID*****。
CredentialScope	见上文，凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文，参与签名的头部信息。此示例取值为 content-type;host;x-tc-action。
Signature	签名值。此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}
```

注意：

请求发送时的 HTTP 头部（Header）和请求体（Payload）必须和签名计算过程中的内容完全一致，否则会返回签名不一致错误。可以通过打印实际请求内容，网络抓包等方式对比排查。

签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚地解释签名过程，下面以实际编程语言为例，将上述的签名过程完整实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

Java

```
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
    private final static String CT_JSON = "application/json; charset=utf-8";

    public static byte[] hmac256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(UTF8));
    }

    public static String sha256Hex(String s) throws Exception {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] d = md.digest(s.getBytes(UTF8));
        return DatatypeConverter.printHexBinary(d).toLowerCase();
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.tencentcloudapi.com";
        String region = "ap-guangzhou";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1551113065";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // 注意时区, 否则容易出错
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ***** 步骤 1: 拼接规范请求串 *****
        String httpRequestMethod = "POST";
```

```
String canonicalUri = "/";
String canonicalQueryString = "";
String canonicalHeaders = "content-type:application/json; charset=utf-8\n"
+ "host:" + host + "\n" + "x-tc-action:" + action.toLowerCase() + "\n";
String signedHeaders = "content-type;host;x-tc-action";

String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}";
String hashedRequestPayload = sha256Hex(payload);
String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

// ***** 步骤 2: 拼接待签名字符串 *****
String credentialScope = date + "/" + service + "/" + "tc3_request";
String hashedCanonicalRequest = sha256Hex(canonicalRequest);
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
System.out.println(stringToSign);

// ***** 步骤 3: 计算签名 *****
byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
byte[] secretService = hmac256(secretDate, service);
byte[] secretSigning = hmac256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
System.out.println(signature);

// ***** 步骤 4: 拼接 Authorization *****
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
.append(" -H \"Authorization: ").append(authorization).append("\")")
.append(" -H \"Content-Type: application/json; charset=utf-8\"")
.append(" -H \"Host: ").append(host).append("\")")
.append(" -H \"X-TC-Action: ").append(action).append("\")")
.append(" -H \"X-TC-Timestamp: ").append(timestamp).append("\")")
.append(" -H \"X-TC-Version: ").append(version).append("\")")
.append(" -H \"X-TC-Region: ").append(region).append("\")")
.append(" -d '").append(payload).append("'");
System.out.println(sb.toString());
}
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcnow().strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Values": [u"未命名"], "Name": "instance-name"}]}

# ***** 步骤 1: 拼接规范请求串 *****
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\nx-tc-action:%s\n" % (ct, host, action.lower())
signed_headers = "content-type;host;x-tc-action"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ***** 步骤 2: 拼接待签名字符串 *****
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)

# ***** 步骤 3: 计算签名 *****
# 计算签名摘要函数
```

```
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()

secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ***** 步骤 4: 拼接 Authorization *****
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"
+ ' -H "Host: ' + host + '"
+ ' -H "X-TC-Action: ' + action + '"
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"
+ ' -H "X-TC-Version: ' + version + '"
+ ' -H "X-TC-Region: ' + region + '"
+ " -d '" + payload + '"')
```

Golang

```
package main

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/hex"
    "fmt"
    "os"
    "strings"
    "time"
)

func sha256hex(s string) string {
    b := sha256.Sum256([]byte(s))
    return hex.EncodeToString(b[:])
}

func hmacsha256(s, key string) string {
    hashed := hmac.New(sha256.New, []byte(key))
    hashed.Write([]byte(s))
    return string(hashed.Sum(nil))
}

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
```

```
secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
host := "cvm.tencentcloudapi.com"
algorithm := "TC3-HMAC-SHA256"
service := "cvm"
version := "2017-03-12"
action := "DescribeInstances"
region := "ap-guangzhou"
//var timestamp int64 = time.Now().Unix()
var timestamp int64 = 1551113065

// step 1: build canonical request string
httpRequestMethod := "POST"
canonicalURI := "/"
canonicalQueryString := ""
canonicalHeaders := fmt.Sprintf("content-type:%s\nhost:%s\nx-tc-action:%s\n",
    "application/json; charset=utf-8", host, strings.ToLower(action))
signedHeaders := "content-type;host;x-tc-action"
payload := `{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}`
hashedRequestPayload := sha256hex(payload)
canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
    httpRequestMethod,
    canonicalURI,
    canonicalQueryString,
    canonicalHeaders,
    signedHeaders,
    hashedRequestPayload)
fmt.Println(canonicalRequest)

// step 2: build string to sign
date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
hashedCanonicalRequest := sha256hex(canonicalRequest)
string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
    algorithm,
    timestamp,
    credentialScope,
    hashedCanonicalRequest)
fmt.Println(string2sign)

// step 3: sign string
secretDate := hmacsha256(date, "TC3"+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
    algorithm,
    secretId,
    credentialScope,
```



```
signedHeaders,  
signature)  
fmt.Println(authorization)  
  
curl := fmt.Sprintf(`curl -X POST https://%s\  
-H "Authorization: %s\  
-H "Content-Type: application/json; charset=utf-8\  
-H "Host: %s" -H "X-TC-Action: %s\  
-H "X-TC-Timestamp: %d\  
-H "X-TC-Version: %s\  
-H "X-TC-Region: %s\  
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)  
fmt.Println(curl)  
}
```

PHP

```
<?php  
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****  
$secretId = getenv("TENCENTCLOUD_SECRET_ID");  
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****  
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");  
$host = "cvm.tencentcloudapi.com";  
$service = "cvm";  
$version = "2017-03-12";  
$action = "DescribeInstances";  
$region = "ap-guangzhou";  
// $timestamp = time();  
$timestamp = 1551113065;  
$algorithm = "TC3-HMAC-SHA256";  
  
// step 1: build canonical request string  
$httpRequestMethod = "POST";  
$canonicalUri = "/";  
$canonicalQueryString = "";  
$canonicalHeaders = implode("\n", [  
"content-type:application/json; charset=utf-8",  
"host:".$host,  
"x-tc-action:".strtolower($action),  
""  
]);  
$signedHeaders = implode(";", [  
"content-type",  
"host",  
"x-tc-action",  
]);  
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]';  
$hashedRequestPayload = hash("SHA256", $payload);  
$canonicalRequest = $httpRequestMethod."\n"  
. $canonicalUri."\n"  
. $canonicalQueryString."\n"  
. $canonicalHeaders."\n"
```

```
.$signedHeaders."\n"
.$shashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

// step 2: build string to sign
$date = gmtime("Y-m-d", $timestamp);
$credentialScope = $date."/".$service."/tc3_request";
$shashedCanonicalRequest = hash("SHA256", $canonicalRequest);
$stringToSign = $algorithm."\n"
.$timestamp."\n"
.$credentialScope."\n"
.$shashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

// step 3: sign string
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;

// step 4: build authorization
$authorization = $algorithm
." Credential=".$secretId."/".$credentialScope
.", SignedHeaders=".$signedHeaders.", Signature=".$signature;
echo $authorization.PHP_EOL;

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"
.' -H "Content-Type: application/json; charset=utf-8"
.' -H "Host: '.$host.'"
.' -H "X-TC-Action: '.$action.'"
.' -H "X-TC-Timestamp: '.$timestamp.'"
.' -H "X-TC-Version: '.$version.'"
.' -H "X-TC-Region: '.$region.'"
." -d '$payload.'"
echo $curl.PHP_EOL;
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'digest'
require 'json'
require 'time'
require 'openssl'

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]
```

```
service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

# ***** 步骤 1: 拼接规范请求串 *****
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
canonical_headers = "content-type:application/json; charset=utf-8\nhost:#{host}\nx-tc-action:#{action.downcase}\n"
signed_headers = 'content-type;host;x-tc-action'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['未命名'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in example, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
  http_request_method,
  canonical_uri,
  canonical_querystring,
  canonical_headers,
  signed_headers,
  hashed_request_payload,
].join("\n")

puts canonical_request

# ***** 步骤 2: 拼接待签名字符串 *****
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
  algorithm,
  timestamp.to_s,
  credential_scope,
  hashed_request_payload,
].join("\n")
puts string_to_sign

# ***** 步骤 3: 计算签名 *****
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature
```

```
# ***** 步骤 4: 拼接 Authorization *****
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signature=#{signature}"
puts authorization

puts 'curl -X POST ' + endpoint \
+ ' -H "Authorization: ' + authorization + '" \
+ ' -H "Content-Type: application/json; charset=utf-8" \
+ ' -H "Host: ' + host + '" \
+ ' -H "X-TC-Action: ' + action + '" \
+ ' -H "X-TC-Timestamp: ' + timestamp.to_s + '" \
+ ' -H "X-TC-Version: ' + version + '" \
+ ' -H "X-TC-Region: ' + region + '" \
+ " -d '" + payload + '"'
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;

public class Application
{
    public static string SHA256Hex(string s)
    {
        using (SHA256 algo = SHA256.Create())
        {
            byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
            StringBuilder builder = new StringBuilder();
            for (int i = 0; i < hashbytes.Length; ++i)
            {
                builder.Append(hashbytes[i].ToString("x2"));
            }
            return builder.ToString();
        }
    }

    public static byte[] HmacSHA256(byte[] key, byte[] msg)
    {
        using (HMACSHA256 mac = new HMACSHA256(key))
        {
            return mac.ComputeHash(msg);
        }
    }

    public static Dictionary<String, String> BuildHeaders(string secretid,
string secretkey, string service, string endpoint, string region,
string action, string version, DateTime date, string requestPayload)
    {
        string datestr = date.ToString("yyyy-MM-dd");
        DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
```

```
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000;
// ***** 步骤 1: 拼接规范请求串 *****
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n"
+ "host:" + endpoint + "\n"
+ "x-tc-action:" + action.ToLower() + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);

// ***** 步骤 2: 拼接待签名字符串 *****
string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "\n"
+ requestTimestamp.ToString() + "\n"
+ credentialScope + "\n"
+ hashedCanonicalRequest;
Console.WriteLine(stringToSign);

// ***** 步骤 3: 计算签名 *****
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);

// ***** 步骤 4: 拼接 Authorization *****
string authorization = algorithm + " "
+ "Credential=" + secretid + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", "
+ "Signature=" + signature;
Console.WriteLine(authorization);

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
```

```
headers.Add("X-TC-Region", region);
return headers;
}
public static void Main(string[] args)
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

    string service = "cvm";
    string endpoint = "cvm.tencentcloudapi.com";
    string region = "ap-guangzhou";
    string action = "DescribeInstances";
    string version = "2017-03-12";

    // 此处由于示例规范的原因, 采用时间戳2019-02-26 00:44:25, 此参数作为示例, 如果在项目中, 您应当使用:
    // DateTime date = DateTime.UtcNow;
    // 注意时区, 建议此时间统一采用UTC时间戳, 否则容易出错
    DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
    string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}\"";

    Dictionary<string, string> headers = BuildHeaders(SECRET_ID, SECRET_KEY, service
, endpoint, region, action, version, date, requestPayload);

    Console.WriteLine("POST https://cvm.tencentcloudapi.com");
    foreach (KeyValuePair<string, string> kv in headers)
    {
        Console.WriteLine(kv.Key + ": " + kv.Value);
    }
    Console.WriteLine();
    Console.WriteLine(requestPayload);
}
}
```

NodeJS

```
const crypto = require('crypto');

function sha256(message, secret = '', encoding) {
    const hmac = crypto.createHmac('sha256', secret)
    return hmac.update(message).digest(encoding)
}

function getHash(message, encoding = 'hex') {
    const hash = crypto.createHash('sha256')
    return hash.update(message).digest(encoding)
}

function getDate(timestamp) {
```

```
const date = new Date(timestamp * 1000)
const year = date.getUTCFullYear()
const month = ('0' + (date.getUTCMonth() + 1)).slice(-2)
const day = ('0' + date.getUTCDate()).slice(-2)
return `${year}-${month}-${day}`
}

function main(){
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

const endpoint = "cvm.tencentcloudapi.com"
const service = "cvm"
const region = "ap-guangzhou"
const action = "DescribeInstances"
const version = "2017-03-12"
//const timestamp = getTime()
const timestamp = 1551113065
//时间处理, 获取世界时间日期
const date = getDate(timestamp)

// ***** 步骤 1: 拼接规范请求串 *****
const payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}"

const hashedRequestPayload = getHash(payload);
const httpRequestMethod = "POST"
const canonicalUri = "/"
const canonicalQueryString = ""
const canonicalHeaders = "content-type:application/json; charset=utf-8\n"
+ "host:" + endpoint + "\n"
+ "x-tc-action:" + action.toLowerCase() + "\n"
const signedHeaders = "content-type;host;x-tc-action"

const canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload
console.log(canonicalRequest)

// ***** 步骤 2: 拼接待签名字符串 *****
const algorithm = "TC3-HMAC-SHA256"
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "\n" +
timestamp + "\n" +
credentialScope + "\n" +
hashedCanonicalRequest
```

```
console.log(stringToSign)

// ***** 步骤 3: 计算签名 *****
const kDate = sha256(date, 'TC3' + SECRET_KEY)
const kService = sha256(service, kDate)
const kSigning = sha256('tc3_request', kService)
const signature = sha256(stringToSign, kSigning, 'hex')
console.log(signature)

// ***** 步骤 4: 拼接 Authorization *****
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
"Signature=" + signature
console.log(authorization)

const curlcmd = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + endpoint + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + '"'
console.log(curlcmd)
}
main()
```

C++

```
#include <algorithm>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

string get_data(int64_t &timestamp)
{
    string utcDate;
    char buff[20] = {0};
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
```



```
utcDate = string(buff);
return utcDate;
}

string int2str(int64_t n)
{
    std::stringstream ss;
    ss << n;
    return ss.str();
}

string sha256Hex(const string &str)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str.c_str(), str.size());
    SHA256_Final(hash, &sha256);
    std::string NewString = "";
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        snprintf(buf, sizeof(buf), "%02x", hash[i]);
        NewString = NewString + buf;
    }
    return NewString;
}

string HmacSha256(const string &key, const string &input)
{
    unsigned char hash[32];

    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
    h = &hmac;
    #else
    h = HMAC_CTX_new();
    #endif

    HMAC_Init_ex(h, &key[0], key.length(), EVP_sha256(), NULL);
    HMAC_Update(h, (unsigned char*)&input[0], input.length());
    unsigned int len = 32;
    HMAC_Final(h, hash, &len);

    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX_cleanup(h);
    #else
    HMAC_CTX_free(h);
    #endif

    std::stringstream ss;
```

```
ss << std::setfill('0');
for (int i = 0; i < len; i++)
{
    ss << hash[i];
}

return (ss.str());
}

string HexEncode(const string &input)
{
    static const char* const lut = "0123456789abcdef";
    size_t len = input.length();

    string output;
    output.reserve(2 * len);
    for (size_t i = 0; i < len; ++i)
    {
        const unsigned char c = input[i];
        output.push_back(lut[c >> 4]);
        output.push_back(lut[c & 15]);
    }
    return output;
}

int main()
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    string SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");

    string service = "cvm";
    string host = "cvm.tencentcloudapi.com";
    string region = "ap-guangzhou";
    string action = "DescribeInstances";
    string version = "2017-03-12";
    int64_t timestamp = 1551113065;
    string date = get_data(timestamp);

    // ***** 步骤 1: 拼接规范请求串 *****
    string httpRequestMethod = "POST";
    string canonicalUri = "/";
    string canonicalQueryString = "";
    string lower = action;
    std::transform(action.begin(), action.end(), lower.begin(), ::tolower);
    string canonicalHeaders = string("content-type:application/json; charset=utf-8\n")
    + "host:" + host + "\n"
    + "x-tc-action:" + lower + "\n";
    string signedHeaders = "content-type;host;x-tc-action";
    string payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}"
};
```

```
string hashedRequestPayload = sha256Hex(payload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
cout << canonicalRequest << endl;

// ***** 步骤 2: 拼接待签名字符串 *****
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
cout << stringToSign << endl;

// ***** 步骤 3: 计算签名 *****
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;

// ***** 步骤 4: 拼接 Authorization *****
string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;

string curlcmd = "curl -X POST https://" + host + "\n"
+ " -H \"Authorization: \" + authorization + "\"\n"
+ " -H \"Content-Type: application/json; charset=utf-8\"\n"
+ " -H \"Host: \" + host + "\"\n"
+ " -H \"X-TC-Action: \" + action + "\"\n"
+ " -H \"X-TC-Timestamp: \" + RequestTimestamp + "\"\n"
+ " -H \"X-TC-Version: \" + version + "\"\n"
+ " -H \"X-TC-Region: \" + region + "\"\n"
+ " -d '" + payload + "'";
cout << curlcmd << endl;
return 0;
};
```

C

```
#include <ctype.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdint.h>
```

```
#include <openssl/sha.h>
#include <openssl/hmac.h>

void get_utc_date(int64_t timestamp, char* utc, int len)
{
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(utc, len, "%Y-%m-%d", &sttime);
}

void sha256_hex(const char* str, char* result)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str, strlen(str));
    SHA256_Final(hash, &sha256);
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        sprintf(buf, sizeof(buf), "%02x", hash[i]);
        strcat(result, buf);
    }
}

void hmac_sha256(const char* key, int key_len,
const char* input, int input_len,
unsigned char* output, unsigned int* output_len)
{
    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
    h = &hmac;
    #else
    h = HMAC_CTX_new();
    #endif

    HMAC_Init_ex(h, key, key_len, EVP_sha256(), NULL);
    HMAC_Update(h, ( unsigned char* )input, input_len);
    HMAC_Final(h, output, output_len);

    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX_cleanup(h);
    #else
    HMAC_CTX_free(h);
    #endif
}

void hex_encode(const char* input, int input_len, char* output)
```

```
{
static const char* const lut = "0123456789abcdef";

char add_out[128] = {0};
char temp[2] = {0};
for (size_t i = 0; i < input_len; ++i)
{
const unsigned char c = input[i];
temp[0] = lut[c >> 4];
strcat(add_out, temp);
temp[0] = lut[c & 15];
strcat(add_out, temp);
}
strncpy(output, add_out, 128);
}

void lowercase(const char * src, char * dst)
{
for (int i = 0; src[i]; i++)
{
dst[i] = tolower(src[i]);
}
}

int main()
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const char* SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");
const char* service = "cvm";
const char* host = "cvm.tencentcloudapi.com";
const char* region = "ap-guangzhou";
const char* action = "DescribeInstances";
const char* version = "2017-03-12";
int64_t timestamp = 1551113065;
char date[20] = {0};
get_utc_date(timestamp, date, sizeof(date));

// ***** 步骤 1: 拼接规范请求串 *****
const char* http_request_method = "POST";
const char* canonical_uri = "/";
const char* canonical_query_string = "";
char canonical_headers[100] = {"content-type:application/json; charset=utf-8\nhost:"};
strcat(canonical_headers, host);
strcat(canonical_headers, "\nx-tc-action:");
char value[100] = {0};
lowercase(action, value);
strcat(canonical_headers, value);
strcat(canonical_headers, "\n");
const char* signed_headers = "content-type;host;x-tc-action";
const char* payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance
```

```
-name\}}]};";
char hashed_request_payload[100] = {0};
sha256_hex(payload, hashed_request_payload);

char canonical_request[256] = {0};
sprintf(canonical_request, "%s\n%s\n%s\n%s\n%s\n%s", http_request_method,
canonical_uri, canonical_query_string, canonical_headers,
signed_headers, hashed_request_payload);
printf("%s\n", canonical_request);

// ***** 步骤 2: 拼接待签名字符串 *****
const char* algorithm = "TC3-HMAC-SHA256";
char request_timestamp[16] = {0};
sprintf(request_timestamp, "%d", timestamp);
char credential_scope[64] = {0};
strcat(credential_scope, date);
sprintf(credential_scope, "%s/%s/tc3_request", date, service);
char hashed_canonical_request[100] = {0};
sha256_hex(canonical_request, hashed_canonical_request);
char string_to_sign[256] = {0};
sprintf(string_to_sign, "%s\n%s\n%s\n%s", algorithm, request_timestamp,
credential_scope, hashed_canonical_request);
printf("%s\n", string_to_sign);

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s%s", "TC3", SECRET_KEY);
unsigned char k_date[64] = {0};
unsigned int output_len = 0;
hmac_sha256(k_key, strlen(k_key), date, strlen(date), k_date, &output_len);
unsigned char k_service[64] = {0};
hmac_sha256(k_date, output_len, service, strlen(service), k_service, &output_len);
unsigned char k_signing[64] = {0};
hmac_sha256(k_service, output_len, "tc3_request", strlen("tc3_request"), k_signing, &output_len);
unsigned char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, output_len, string_to_sign, strlen(string_to_sign), k_hmac_sha_sign, &output_len);

char signature[128] = {0};
hex_encode(k_hmac_sha_sign, output_len, signature);
printf("%s\n", signature);

// ***** 步骤 4: 拼接 Authorization *****
char authorization[512] = {0};
sprintf(authorization, "%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm, SECRET_ID, credential_scope, signed_headers, signature);
printf("%s\n", authorization);

char curlcmd[10240] = {0};
sprintf(curlcmd, "curl -X POST https://%s\n \
-H \"Authorization: %s\"\n \
-H \"Content-Type: application/json; charset=utf-8\"\n \
-H \"Host: %s\"\n \
```

```
-H \"X-TC-Action: %s\"\\n \\  
-H \"X-TC-Timestamp: %s\"\\n \\  
-H \"X-TC-Version: %s\"\\n \\  
-H \"X-TC-Region: %s\"\\n \\  
-d \"%s\",  
host, authorization, host, action, request_timestamp, version, region, payload);  
printf(\"%s\\n\", curlcmd);  
return 0;  
}
```

其他语言

- Lua: [GitHub](#)
- Swift: [GitHub](#)
- Dart: [GitHub](#)
- Shell(Bash): [GitHub](#)

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

错误码	错误描述
AuthFailure.SignatureExpire	签名过期。Timestamp 与服务器接收到请求的时间相差不得超过五分钟。
AuthFailure.SecretIdNotFound	密钥不存在。请到控制台查看密钥是否被禁用，是否少复制了字符或者多了字符。
AuthFailure.SignatureFailure	签名错误。可能是签名计算错误，或者签名与实际发送的内容不符合，也有可能是密钥 SecretKey 错误导致的。
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

签名方法

最近更新时间：2024-12-25 01:10:37

签名方法 v1 简单易用，但是功能和安全性都不如签名方法 v3，推荐使用签名方法 v3。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

推荐使用 API Explorer

<> 点击调试

您可以通过 API Explorer 的 [签名串生成] 模块查看每个接口签名的生成过程。

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息 (Signature) 以验证请求者身份。

签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往 [云API密钥页面](#) 申请，否则无法调用云 API 接口。

1. 申请安全凭证

在第一次使用云 API 之前，请前往 [云 API 密钥页面](#) 申请安全凭证。

安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录 [腾讯云管理中心控制台](#)。
2. 前往 [云 API 密钥](#) 的控制台页面
3. 在 [云 API 密钥](#) 页面，单击 [新建密钥] 即可以创建一对 SecretId/SecretKey。

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

有了安全凭证 SecretId 和 SecretKey 后，就可以生成签名串了。以下是使用签名方法 v1 生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKID*****
- SecretKey: *****

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表 (DescribeInstances) 请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥 ID	AKID*****
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例 ID	ins-09dx96dg

参数名称	中文	参数值
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数。

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 PHP 中的 ksort 函数。上述示例参数的排序结果如下：

```
{
  'Action' : 'DescribeInstances',
  'InstanceIds.0' : 'ins-09dx96dg',
  'Limit' : 20,
  'Nonce' : 11886,
  'Offset' : 0,
  'Region' : 'ap-guangzhou',
  'SecretId' : 'AKID*****',
  'Timestamp' : 1465185768,
  'Version' : '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。

将把上一步排序好的请求参数格式化“参数名称=参数值”的形式，如对 Action 参数，其参数名称为 "Action"，参数值为 "DescribeInstances"，因此格式化后即为 Action=DescribeInstances。

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机: 查看实例列表(DescribeInstances)的请求域名为: cvm.tencentcloudapi.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 /。
4. 请求字符串: 即上一步生成的请求字符串。

签名原串的拼接规则为：请求方法 + 请求主机 + 请求路径 + ? + 请求字符串。

示例的拼接结果为：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。

首先使用 HMAC-SHA1 算法对上一步中获得的**签名原文字符串**进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例：

```
$secretKey = '*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 7RAM2xfNMO9EiVTNmPg06MRnCvQ=，最终得到的签名串请求参数（Signature）为：7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先用 UTF-8 进行编码。

注意：有些编程语言的库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

注意：其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12。`

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
        // 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
    }
}
```

```
return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
    StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
    // 实际请求的url中对参数顺序没有要求
    for (String k : params.keySet()) {
        // 需要对请求串进行urlencode, 由于key都是英文字母, 故此处仅对其value进行urlencode
        url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
    }
    return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
    // 实际调用时应当使用随机数, 例如: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
    params.put("Nonce", 11886); // 公共参数
    // 实际调用时应当使用系统当前时间, 例如: params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // 公共参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 公共参数
    params.put("Action", "DescribeInstances"); // 公共参数
    params.put("Version", "2017-03-12"); // 公共参数
    params.put("Region", "ap-guangzhou"); // 公共参数
    params.put("Limit", 20); // 业务参数
    params.put("Offset", 0); // 业务参数
    params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "HmacSHA1")); //
    公共参数
    System.out.println(getUrl(params));
}
}
```

Python

注意: 如果是在 Python 2 环境中运行, 需要先安装 requests 依赖包: `pip install requests`。

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import os
import time

import requests

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.getenv("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.getenv("TENCENTCLOUD_SECRET_KEY")

def get_string_to_sign(method, endpoint, params):
```

```
s = method + endpoint + "?"
query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceIds.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

Golang

```
package main

import (
    "bytes"
    "crypto/hmac"
    "crypto/sha1"
    "encoding/base64"
    "fmt"
    "os"
    "sort"
    "strconv"
)

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    params := map[string]string{
        "Nonce": "11886",
        "Timestamp": strconv.Itoa(1465185768),
        "Region": "ap-guangzhou",
```

```
"SecretId": secretId,
"Version": "2017-03-12",
"Action": "DescribeInstances",
"InstanceIds.0": "ins-09dx96dg",
"Limit": strconv.Itoa(20),
"Offset": strconv.Itoa(0),
}

var buf bytes.Buffer
buf.WriteString("GET")
buf.WriteString("cvm.tencentcloudapi.com")
buf.WriteString("/")
buf.WriteString("?")

// sort keys by ascii asc order
keys := make([]string, 0, len(params))
for k, _ := range params {
    keys = append(keys, k)
}
sort.Strings(keys)

for i := range keys {
    k := keys[i]
    buf.WriteString(k)
    buf.WriteString("=")
    buf.WriteString(params[k])
    buf.WriteString("&")
}
buf.Truncate(buf.Len() - 1)

hashed := hmac.New(sha1.New, []byte(secretKey))
hashed.Write(buf.Bytes())

fmt.Println(base64.StdEncoding.EncodeToString(hashed.Sum(nil)))
}
```

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceIds.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;
```

```
ksort($param);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $param as $key => $value ) {
    $signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
// need to install and enable curl extension in php.ini
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'time'
require 'openssl'
require 'base64'

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceIds.0' => 'ins-09dx96dg',
  'Limit' => 20,
  'Nonce' => 11886,
  'Offset' => 0,
  'Region' => 'ap-guangzhou',
  'SecretId' => secret_id,
  'Timestamp' => 1465185768, # Time.now.to_i
  'Version' => '2017-03-12',
}
sign = method + endpoint + '?'
params = []
data.sort.each do |item|
  params << "#{item[0]}=#{item[1]}"
end
end
```

```
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sign))
puts data['Signature']

# require 'net/http'
# uri = URI('https://' + endpoint)
# uri.query = URI.encode_www_form(data)
# p uri
# res = Net::HTTP.get_response(uri)
# puts res.body
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Security.Cryptography;
using System.Text;

public class Application {
    public static string Sign(string signKey, string secret)
    {
        string signRet = string.Empty;
        using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
        {
            byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
            signRet = Convert.ToBase64String(hash);
        }
        return signRet;
    }

    public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
    {
        string retStr = "";
        retStr += requestMethod;
        retStr += requestHost;
        retStr += requestPath;
        retStr += "?";
        string v = "";
        foreach (string key in requestParams.Keys)
        {
            v += string.Format("{0}={1}&", key, requestParams[key]);
        }
        retStr += v.TrimEnd('&');
        return retStr;
    }

    public static void Main(string[] args)
    {
        // 密钥参数
        // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
```



```
string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
double RequestTimestamp = 1465185768; // 时间戳 2019-02-26 00:44:25,此参数作为示例,以实际为准
// long timestamp = ToTimestamp() / 1000;
// string requestTimestamp = timestamp.ToString();
Dictionary<string, string> param = new Dictionary<string, string>();
param.Add("Limit", "20");
param.Add("Offset", "0");
param.Add("InstanceIds.0", "ins-09dx96dg");
param.Add("Action", action);
param.Add("Nonce", "11886");
// param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

param.Add("Timestamp", RequestTimestamp.ToString());
param.Add("Version", version);

param.Add("SecretId", SECRET_ID);
param.Add("Region", region);
SortedDictionary<string, string> headers = new SortedDictionary<string, string>(param, StringComparer.Ordinal);
string sigInParameter = MakeSignPlainText(headers, "GET", endpoint, "/");
string sigOutParameter = Sign(SECRET_KEY, sigInParameter);
Console.WriteLine(sigOutParameter);
}
}
```

NodeJS

```
const crypto = require('crypto');

function get_req_url(params, endpoint){
    params['Signature'] = encodeURIComponent(params['Signature']);
    const url_strParam = sort_params(params)
    return "https://" + endpoint + "/" + url_strParam.slice(1);
}

function formatSignString(reqMethod, endpoint, path, strParam){
    let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
    return strSign;
}

function sha1(secretKey, strsign){
    let signMethodMap = {'HmacSHA1': "sha1"};
    let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
    return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

function sort_params(params){
```

```
let strParam = "";
let keys = Object.keys(params);
keys.sort();
for (let k in keys) {
  //k = k.replace(/_/g, '.');
  strParam += ("%&" + keys[k] + "=" + params[keys[k]]);
}
return strParam
}

function main(){
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

const endpoint = "cvm.tencentcloudapi.com"
const Region = "ap-guangzhou"
const Version = "2017-03-12"
const Action = "DescribeInstances"
const Timestamp = 1465185768 // 时间戳 2016-06-06 12:02:48, 此参数作为示例, 以实际为准
// const Timestamp = Math.round(Date.now() / 1000)
const Nonce = 11886 // 随机正整数
//const nonce = Math.round(Math.random() * 65535)

let params = {};
params['Action'] = Action;
params['InstanceIds.0'] = 'ins-09dx96dg';
params['Limit'] = 20;
params['Offset'] = 0;
params['Nonce'] = Nonce;
params['Region'] = Region;
params['SecretId'] = SECRET_ID;
params['Timestamp'] = Timestamp;
params['Version'] = Version;

// 1. 对参数排序, 并拼接请求字符串
strParam = sort_params(params)

// 2. 拼接签名原字符串
const reqMethod = "GET";
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
// console.log(strSign)

// 3. 生成签名串
params['Signature'] = sha1(SECRET_KEY, strSign)
console.log(params['Signature'])

// 4. 进行url编码并拼接请求url
// const req_url = get_req_url(params, endpoint)
// console.log(params['Signature'])
```

```
// console.log(req_url)
}  
main()
```

返回结果

最近更新时间：2024-03-12 01:09:23

云 API 3.0 接口默认返回 JSON 数据，返回非 JSON 格式的接口会在文档中做出说明。返回 JSON 数据时最大限制为 50 MB，如果返回的数据超过最大限制，请求会失败并返回内部错误。请根据接口文档中给出的过滤功能（例如时间范围）或者分页功能，控制返回数据不要过大。

注意：目前只要请求被服务端正常处理了，响应的 HTTP 状态码均为 200。例如返回的消息体里的错误码是签名失败，但 HTTP 状态码是 200，而不是 401。

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系 [腾讯云客服](#) 或 [提交工单](#)，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0，InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系 [腾讯云客服](#) 或 [提交工单](#)，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码。完整的错误码列表请参考本产品“API 文档”目录下的“错误码”页面。

参数类型

最近更新时间：2022-08-10 06:14:16

目前腾讯云 API 3.0 输入参数和输出参数支持如下几种数据格式：

- String: 字符串。
- Integer: 整型，上限为无符号64位整数。SDK 3.0 不同编程语言支持的类型有所差异，建议以所使用编程语言的最大的整型定义，例如 Golang 的 `uint64`。
- Boolean: 布尔型。
- Float: 浮点型。
- Double: 双精度浮点型。
- Date: 字符串，日期格式。例如：2022-01-01。
- Timestamp: 字符串，时间格式。例如：2022-01-01 00:00:00。
- Timestamp ISO8601: ISO 8601 是由国际标准化组织（International Organization for Standardization, ISO）发布的关于日期和时间格式的国际标准，对应国标《[GB/T 7408-2005数据元和交换格式信息交换日期和时间表示法](#)》。建议以所使用编程语言的标准库进行格式解析。例如：2022-01-01T00:00:00+08:00。
- Binary: 二进制内容，需要以特定协议请求和解析。

伸缩组相关接口

修改期望实例数

最近更新时间：2025-05-19 01:09:32

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ModifyDesiredCapacity）用于修改指定伸缩组的期望实例数

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyDesiredCapacity。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-nvnlpbb8
DesiredCapacity	是	Integer	期望实例数，取值范围 [0,2000]，同时需满足 $MaxSize \geq DesiredCapacity \geq MinSize$ 。 示例值：2
MinSize	否	Integer	最小实例数，取值范围为 [0,2000]，同时需满足 $MaxSize \geq DesiredCapacity \geq MinSize$ 。 示例值：0
MaxSize	否	Integer	最大实例数，取值范围为 [0,2000]，同时需满足 $MaxSize \geq DesiredCapacity \geq MinSize$ 。 示例值：20

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改伸缩组期望实例数

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
```

```
Content-Type: application/json
X-TC-Action: ModifyDesiredCapacity
<公共请求参数>

{
  "AutoScalingGroupId": "asg-nvnlpbb8",
  "DesiredCapacity": 2
}
```

输出示例

```
{
  "Response": {
    "RequestId": "2f7c0f11-edfd-4598-a5f6-fb5c10cc9d8e"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.AssertDesiredCapacityFailed	声明的原有期望实例数和实际原有期望实例数不一致。
InvalidParameterValue.BaseCapacityTooLarge	指定的基础容量过大，需小于等于最大实例数。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。

错误码	描述
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.Size	伸缩组最大数量、最小数量、期望实例数取值不合法。
LimitExceeded.DesiredCapacityLimitExceeded	期望实例数超出限制。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceUnavailable.AutoScalingGroupAbnormalStatus	伸缩组状态异常。
ResourceUnavailable.AutoScalingGroupDisabled	伸缩组被停用。

根据实例创建启动配置及伸缩组

最近更新时间：2025-05-19 01:09:32

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（CreateAutoScalingGroupFromInstance）用于根据实例创建启动配置及伸缩组。

说明：根据按包年包月计费的实例所创建的伸缩组，其扩容的实例为按量计费实例。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateAutoScalingGroupFromInstance。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupName	是	String	伸缩组名称，在您账号中必须唯一。名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点，最大长度不能超55个字节。 示例值：asgname
InstanceId	是	String	实例ID。可通过登录 控制台 或调用接口 DescribeInstances ，取返回信息中的 InstanceId 获取实例ID。 示例值：ins-aaaabbbb
MinSize	是	Integer	最小实例数，取值范围为 [0,2000]，同时需满足 $MaxSize \geq DesiredCapacity \geq MinSize$ 。 示例值：1
MaxSize	是	Integer	最大实例数，取值范围为 [0,2000]，同时需满足 $MaxSize \geq DesiredCapacity \geq MinSize$ 。 示例值：1
DesiredCapacity	否	Integer	期望实例数，取值范围 [0,2000]，默认值等于当前 MinSize，同时需满足 $MaxSize \geq DesiredCapacity \geq MinSize$ 。 示例值：1
InheritInstanceTag	否	Boolean	是否继承实例标签，默认值为False 示例值：False

3. 输出参数

参数名称	类型	描述
AutoScalingGroupId	String	伸缩组ID 示例值：asg-aaaabbbb

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 根据实例创建启动配置及伸缩组

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateAutoScalingGroupFromInstance
<公共请求参数>

{
  "DesiredCapacity": 0,
  "InstanceId": "ins-19a14o9y",
  "AutoScalingGroupName": "as-test",
  "MinSize": 0,
  "MaxSize": 1,
  "InheritInstanceTag": false
}
```

输出示例

```
{
  "Response": {
    "AutoScalingGroupId": "asg-cqatht5b",
    "RequestId": "19c10733-d0e8-4f58-ac82-e1b1affb0bbb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AccountQualificationRestrictions	该请求账户未通过资格审计。
CallCvmError	CVM接口调用失败。
InternalError	内部错误。
InternalError.CallVpcError	VPC接口调用失败。
InternalError.CalleeError	调用其他服务异常。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameterValue.CvmConfigurationError	CVM参数校验异常。
InvalidParameterValue.CvmError	CVM参数校验异常。
InvalidParameterValue.DuplicatedSubnet	指定的子网重复。
InvalidParameterValue.ForwardLb	在应当指定应用型负载均衡器的参数中，错误地指定了一个非应用型的负载均衡器。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.LaunchConfigurationNameDuplicated	启动配置名称重复。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.Size	伸缩组最大数量、最小数量、期望实例数取值不合法。
InvalidParameterValue.TooLong	取值过多。
LimitExceeded.AutoScalingGroupLimitExceeded	伸缩组数量超过限制。
LimitExceeded.DesiredCapacityLimitExceeded	期望实例数超出限制。
LimitExceeded.LaunchConfigurationQuotaNotEnough	启动配置配额不足。
LimitExceeded.MaxSizeLimitExceeded	最大实例数大于限制。
LimitExceeded.MinSizeLimitExceeded	最小实例数低于限制。
MissingParameter	缺少参数错误。
ResourceNotFound.InstancesNotFound	指定的实例不存在。
ResourceUnavailable.LaunchConfigurationStatusAbnormal	启动配置状态异常。
ResourceUnavailable.ProjectInconsistent	项目不一致。
ResourceUnavailable.StoppedInstanceNotAllowAttach	关机实例不允许添加到伸缩组。

启用伸缩组

最近更新时间：2025-05-16 01:09:34

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（EnableAutoScalingGroup）用于启用指定伸缩组。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：EnableAutoScalingGroup。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-nvnlpbb8

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 启用指定伸缩组

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: EnableAutoScalingGroup
<公共请求参数>

{
  "AutoScalingGroupId": "asg-nvnlpbb8"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "a1c937b5-5f0b-4bf3-bdfb-bb859aceda6f"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceUnavailable.AutoScalingGroupInRefreshActivity	伸缩组处于另一个实例刷新活动当中。

停用伸缩组

最近更新时间：2025-05-16 01:09:34

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DisableAutoScalingGroup）用于停用指定伸缩组。

- 停用伸缩组后，自动触发的伸缩活动不再进行，包括：
 - 告警策略触发的伸缩活动
 - 匹配期望实例数的伸缩活动
 - 不健康实例替换活动
 - 定时任务
- 停用伸缩组后，手动触发的伸缩活动允许进行，包括：
 - 指定数量扩容实例（ScaleOutInstances）
 - 指定数量缩容实例（ScaleInInstances）
 - 从伸缩组中移出 CVM 实例（DetachInstances）
 - 从伸缩组中删除 CVM 实例（RemoveInstances）
 - 添加 CVM 实例到伸缩组（AttachInstances）
 - 关闭伸缩组内 CVM 实例（StopAutoScalingInstances）
 - 开启伸缩组内 CVM 实例（StartAutoScalingInstances）

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DisableAutoScalingGroup。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-nvnlpbb8

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 停用指定伸缩组

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DisableAutoScalingGroup
<公共请求参数>

{
  "AutoScalingGroupId": "asg-nvnlpbb8"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "aa9ead18-44c8-42a8-a68b-ac6206969c2c"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。

错误码	描述
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。

修改伸缩组的负载均衡器

最近更新时间：2025-05-16 01:09:33

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ModifyLoadBalancers）用于修改伸缩组的负载均衡器。

- 本接口用于为伸缩组指定新的负载均衡器配置，采用完全覆盖风格，无论之前配置如何，统一按照接口参数配置为新的负载均衡器。
- 如果要为伸缩组清空负载均衡器，则在调用本接口时仅指定伸缩组ID，不指定具体负载均衡器。
- 本接口会立即修改伸缩组的负载均衡器，并生成一个伸缩活动，异步修改存量实例的负载均衡器。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyLoadBalancers。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-12wjuh0s
LoadBalancerIds.N	否	Array of String	传统负载均衡器ID列表，目前长度上限为20，LoadBalancerIds 和 ForwardLoadBalancers 二者同时最多只能指定一个。可以通过 DescribeLoadBalancers 接口获取。 示例值：["lb-crhgatrf"]
ForwardLoadBalancers.N	否	Array of ForwardLoadBalancer	负载均衡器列表，目前长度上限为100，LoadBalancerIds 和 ForwardLoadBalancers 二者同时最多只能指定一个。可以通过 DescribeLoadBalancers 接口获取。 示例值：[{"LoadBalancerId": "lb-af8li2kn", "ListenerId": "lb-akw1lgo7", "TargetAttributes": [{"Port": 80, "Weight": 10}]}]
LoadBalancersCheckPolicy	否	String	负载均衡器校验策略，取值包括 ALL 和 DIFF，默认取值为 ALL。 <ul style="list-style-type: none">• ALL，所有负载均衡器都合法则通过校验，否则校验报错。• DIFF，仅校验负载均衡器参数中实际变化的部分，如果合法则通过校验，否则校验报错。 示例值：ALL

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-67izy66g

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 清空伸缩组的负载均衡器

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyLoadBalancers
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s"
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-rp63a5q8",
    "RequestId": "7de5a82f-b781-4302-b723-e7a879c20767"
  }
}
```

示例2 将伸缩组的负载均衡器修改为传统负载均衡器lb-crhgatrf

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyLoadBalancers
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "LoadBalancerIds": [
    "lb-crhgatrf"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-67izy66g",
```

```
"RequestId": "bd3c91e8-3051-4c02-ac58-54d47b9c9d63"
}
```

示例3 将伸缩组的负载均衡器修改为应用型负载均衡器lb-23aejgcv，监听器为lbl-ncw704sn

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyLoadBalancers
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "ForwardLoadBalancers": [
    {
      "TargetAttributes": [
        {
          "Port": 8080,
          "Weight": 10
        }
      ],
      "Region": "ap-guangzhou",
      "LocationId": "loc-13hmaev9",
      "ListenerId": "lbl-ncw704sn",
      "LoadBalancerId": "lb-23aejgcv"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-9asddelc",
    "RequestId": "8d78668d-61eb-456d-855b-f34f91371089"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalServerError.CallLbError	LB 接口调用失败。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameterValue.ClassicLb	在应当指定传统型负载均衡器的参数中，错误地指定了一个非传统型的负载均衡器。
InvalidParameterValue.DuplicatedForwardLb	提供的应用型负载均衡器重复。
InvalidParameterValue.ForwardLb	在应当指定应用型负载均衡器的参数中，错误地指定了一个非应用型的负载均衡器。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidClbRegion	为CLB指定的地域不合法。
InvalidParameterValue.LbProjectInconsistent	负载均衡器项目不一致。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.TargetPortDuplicated	负载均衡器四层监听器的后端端口重复。
MissingParameter.InScenario	在特定场景下缺少参数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.ListenerNotFound	指定的Listener不存在。
ResourceNotFound.LoadBalancerNotFound	找不到指定负载均衡器。
ResourceNotFound.LocationNotFound	指定的Location不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.LbBackendRegionInconsistent	CLB实例的后端地域与AS服务所在地域不一致。
ResourceUnavailable.LbVpcInconsistent	负载均衡器VPC与伸缩组不一致。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

修改负载均衡器目标规则属性

最近更新时间：2025-05-16 01:09:33

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ModifyLoadBalancerTargetAttributes）用于修改伸缩组内负载均衡器的目标规则属性。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyLoadBalancerTargetAttributes。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-12wjuh0s
ForwardLoadBalancers.N	是	Array of ForwardLoadBalancer	需修改目标规则属性的负载均衡器列表，列表长度上限为100。可以通过 DescribeLoadBalancers 接口获取。

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-67izy66g
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改应用型负载均衡器lb-23aejgcv（监听器为lb-ncw704sn，转发路径为loc-l3hmaev9）的目标规则属性，将端口修改为8080，权重修改为20。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyLoadBalancerTargetAttributes
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "ForwardLoadBalancers": [
    {
```

```
"TargetAttributes": [
  {
    "Port": 8080,
    "Weight": 20
  }
],
"Region": "ap-guangzhou",
"LocationId": "loc-13hmaev9",
"ListenerId": "lbl-ncw704sn",
"LoadBalancerId": "lb-23aejgcv"
}
]
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-9asddelc",
    "RequestId": "8d78668d-61eb-456d-855b-f34f91371089"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。

错误码	描述
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameter.LoadBalancerNotInAutoScalingGroup	指定的负载均衡器在当前伸缩组中没有找到。
InvalidParameterValue.ClassicLb	在应当指定传统型负载均衡器的参数中，错误地指定了一个非传统型的负载均衡器。
InvalidParameterValue.ForwardLb	在应当指定应用型负载均衡器的参数中，错误地指定了一个非应用型的负载均衡器。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidClbRegion	为CLB指定的地域不合法。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.TargetPortDuplicated	负载均衡器四层监听器的后端端口重复。
LimitExceeded.AfterAttachLbLimitExceeded	绑定指定的负载均衡器后，伸缩组绑定的负载均衡器总数超过了最大值。
MissingParameter.InScenario	在特定场景下缺少参数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.ListenerNotFound	指定的Listener不存在。
ResourceNotFound.LoadBalancerNotFound	找不到指定负载均衡器。
ResourceNotFound.LoadBalancerNotInAutoScalingGroup	指定的负载均衡器在当前伸缩组中没有找到。
ResourceNotFound.LocationNotFound	指定的Location不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。
ResourceUnavailable.LbBackendRegionInconsistent	CLB实例的后端地域与AS服务所在地域不一致。
ResourceUnavailable.LbProjectInconsistent	负载均衡器项目不一致。
ResourceUnavailable.LbVpcInconsistent	负载均衡器VPC与伸缩组不一致。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

移出负载均衡器

最近更新时间：2025-05-16 01:09:34

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DetachLoadBalancers）用于从伸缩组移出负载均衡器，本接口不会销毁负载均衡器。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DetachLoadBalancers。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-12wjuh0s
LoadBalancerIds.N	否	Array of String	传统负载均衡器ID列表，列表长度上限为20，LoadBalancerIds 和 ForwardLoadBalancerIdentifications 二者同时最多只能指定一个。可以通过 DescribeLoadBalancers 接口获取。 示例值：["lb-crhgatrf"]
ForwardLoadBalancerIdentifications.N	否	Array of ForwardLoadBalancerIdentification	负载均衡器标识信息列表，列表长度上限为100，LoadBalancerIds 和 ForwardLoadBalancerIdentifications 二者同时最多只能指定一个。可以通过 DescribeLoadBalancers 接口获取。 示例值：[{"ListenerId": "lbl-mqnrswr0", "LoadBalancerId": "lb-k7cs98bu", "LocationId": "loc-ivbajssu"}]

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-67izy66g
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 将传统负载均衡器lb-crhgatrf从伸缩组中移出

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DetachLoadBalancers
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "LoadBalancerIds": [
    "lb-crhgatrf"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-67izy66g",
    "RequestId": "bd3c91e8-3051-4c02-ac58-54d47b9c9d63"
  }
}
```

示例2 将应用型负载均衡器lb-23aejgcv（监听器为lbl-ncw704sn，转发路径为loc-l3hmaev9）从伸缩组中移出

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DetachLoadBalancers
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "ForwardLoadBalancerIdentifications": [
    {
      "LocationId": "loc-l3hmaev9",
      "ListenerId": "lbl-ncw704sn",
      "LoadBalancerId": "lb-23aejgcv"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-9asddelc",
    "RequestId": "8d78668d-61eb-456d-855b-f34f91371089"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameter.LoadBalancerNotInAutoScalingGroup	指定的负载均衡器在当前伸缩组中没有找到。
InvalidParameter.MustOneParameter	参数缺失，两种参数之中必须指定其中一个。
InvalidParameterValue.ClassicLb	在应当指定传统型负载均衡器的参数中，错误地指定了一个非传统型的负载均衡器。
InvalidParameterValue.ForwardLb	在应当指定应用型负载均衡器的参数中，错误地指定了一个非应用型的负载均衡器。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。

错误码	描述
InvalidParameterValue.InvalidClbRegion	为CLB指定的地域不合法。
LimitExceeded.AfterAttachLbLimitExceeded	绑定指定的负载均衡器后，伸缩组绑定的负载均衡器总数超过了最大值。
MissingParameter.InScenario	在特定场景下缺少参数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.ListenerNotFound	指定的Listener不存在。
ResourceNotFound.LoadBalancerNotFound	找不到指定负载均衡器。
ResourceNotFound.LoadBalancerNotInAutoScalingGroup	指定的负载均衡器在当前伸缩组中没有找到。
ResourceNotFound.LocationNotFound	指定的Location不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.LbBackendRegionInconsistent	CLB实例的后端地域与AS服务所在地域不一致。
ResourceUnavailable.LbProjectInconsistent	负载均衡器项目不一致。
ResourceUnavailable.LbVpcInconsistent	负载均衡器VPC与伸缩组不一致。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

添加负载均衡器

最近更新时间：2025-05-16 01:09:36

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

此接口（AttachLoadBalancers）用于将负载均衡器添加到伸缩组。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：AttachLoadBalancers。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-12wjuh0s
LoadBalancerIds.N	否	Array of String	传统型负载均衡器ID列表，每个伸缩组绑定传统型负载均衡器数量上限为20，LoadBalancerIds 和 ForwardLoadBalancers 二者同时最多只能指定一个。可以通过 DescribeLoadBalancers 接口获取。 示例值：["lb-crhgatrf"]
ForwardLoadBalancers.N	否	Array of ForwardLoadBalancer	负载均衡器列表，每个伸缩组绑定应用型负载均衡器数量上限为100，LoadBalancerIds 和 ForwardLoadBalancers 二者同时最多只能指定一个。可以通过 DescribeLoadBalancers 接口获取。 示例值：[{"LoadBalancerId": "lb-af8li2kn", "ListenerId": "lbi-akw1lgo7", "TargetAttributes": [{"Port": 80, "Weight": 10}]}]

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-67izy66g
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 将传统负载均衡器lb-crhgatrf添加到伸缩组

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: AttachLoadBalancers
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "LoadBalancerIds": [
    "lb-crhgatrf"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-67lzy66g",
    "RequestId": "bd3c91e8-3051-4c02-ac58-54d47b9c9d63"
  }
}
```

示例2 将应用型负载均衡lb-23aejgcv（监听器为lbl-ncw704sn，转发路径为loc-l3hmaev9）添加到伸缩组

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: AttachLoadBalancers
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "ForwardLoadBalancers": [
    {
      "TargetAttributes": [
        {
          "Port": "8080",
          "Weight": "10"
        }
      ],
      "Region": "ap-guangzhou",
      "LocationId": "loc-l3hmaev9",
      "ListenerId": "lbl-ncw704sn",
      "LoadBalancerId": "lb-23aejgcv"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-9asddelc",
    "RequestId": "8d78668d-61eb-456d-855b-f34f91371089"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalServerError.CallLbError	LB 接口调用失败。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameter.MustOneParameter	参数缺失，两种参数之中必须指定其中一个。
InvalidParameterValue.ClassicLb	在应当指定传统型负载均衡器的参数中，错误地指定了一个非传统型的负载均衡器。
InvalidParameterValue.DuplicatedForwardLb	提供的应用型负载均衡器重复。
InvalidParameterValue.ForwardLb	在应当指定应用型负载均衡器的参数中，错误地指定了一个非应用型的负载均衡器。

错误码	描述
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidClbRegion	为CLB指定的地域不合法。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.ListenerTargetTypeNotSupported	不支持目标组类型的监听器。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.TargetPortDuplicated	负载均衡器四层监听器的后端端口重复。
LimitExceeded.AfterAttachLbLimitExceeded	绑定指定的负载均衡器后，伸缩组绑定的负载均衡器总数超过了最大值。
MissingParameter.InScenario	在特定场景下缺少参数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.ListenerNotFound	指定的Listener不存在。
ResourceNotFound.LoadBalancerNotFound	找不到指定负载均衡器。
ResourceNotFound.LocationNotFound	指定的Location不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.LbBackendRegionInconsistent	CLB实例的后端地域与AS服务所在地域不一致。
ResourceUnavailable.LbProjectInconsistent	负载均衡器项目不一致。
ResourceUnavailable.LbVpcInconsistent	负载均衡器VPC与伸缩组不一致。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

查询伸缩活动

最近更新时间：2025-06-13 01:09:11

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeAutoScalingActivities）用于查询伸缩组的伸缩活动记录。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeAutoScalingActivities。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ActivityIds.N	否	Array of String	按照一个或者多个伸缩活动ID查询。伸缩活动ID形如：asa-512ejpfo。每次请求的上限为100。参数不支持同时指定ActivityIds和Filters。 示例值：["asa-i8d6y5r4"]
Filters.N	否	Array of Filter	过滤条件。过滤字段取值范围如下： <ul style="list-style-type: none">• auto-scaling-group-id<ul style="list-style-type: none">◦ 按照【伸缩组ID】进行过滤。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。◦ 类型：String◦ 必选：否◦ 示例值：asg-kiju7yt5• activity-status-code<ul style="list-style-type: none">◦ 按照【伸缩活动状态】进行过滤◦ 类型：String◦ 必选：否◦ 可选项：<ul style="list-style-type: none">▪ INIT：初始化中▪ RUNNING：运行中▪ SUCCESSFUL：活动成功▪ PARTIALLY_SUCCESSFUL：活动部分成功▪ FAILED：活动失败▪ CANCELLED：活动取消• activity-type<ul style="list-style-type: none">◦ 按照【伸缩活动类型】进行过滤◦ 类型：String◦ 必选：否◦ 可选项：<ul style="list-style-type: none">▪ SCALE_OUT：扩容活动▪ SCALE_IN：缩容活动▪ ATTACH_INSTANCES：添加实例▪ REMOVE_INSTANCES：销毁实例▪ DETACH_INSTANCES：移出实例

参数名称	必选	类型	描述
			<ul style="list-style-type: none"> ▪ TERMINATE_INSTANCES_UNEXPECTEDLY: 实例在CVM控制台被销毁 ▪ REPLACE_UNHEALTHY_INSTANCE: 替换不健康实例 ▪ START_INSTANCES: 开启实例 ▪ STOP_INSTANCES: 关闭实例 ▪ INVOKE_COMMAND: 执行命令 <ul style="list-style-type: none"> • activity-id <ul style="list-style-type: none"> ◦ 按照 [伸缩活动ID] 进行过滤。可通过登录 控制台 获取伸缩活动ID。 ◦ 类型: String ◦ 必选: 否 ◦ 示例值: asa-hy6tr4ed <p>每次请求的Filters的上限为10, Filter.Values的上限为5。参数不支持同时指定ActivityIds和Filters。 示例值: [{"Name": "auto-scaling-group-id", "Values": ["asg-mgmeawez"]}]]</p>
Limit	否	Integer	返回数量, 默认为20, 最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值: 100
Offset	否	Integer	偏移量, 默认为0。关于Offset的更进一步介绍请参考 API 简介 中的相关小节。 示例值: 0
StartTime	否	Timestamp ISO8601	伸缩活动最早的开始时间, 如果指定了ActivityIds, 此参数将被忽略。取值为UTC时间, 按照ISO8601标准, 格式: YYYY-MM-DDThh:mm:ssZ。 示例值: 2024-11-11T15:29:59Z
EndTime	否	Timestamp ISO8601	伸缩活动最晚的结束时间, 如果指定了ActivityIds, 此参数将被忽略。取值为UTC时间, 按照ISO8601标准, 格式: YYYY-MM-DDThh:mm:ssZ。 示例值: 2024-11-11T15:29:59Z

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的伸缩活动数量。 示例值: 1
ActivitySet	Array of Activity	符合条件的伸缩活动信息集合。
RequestId	String	唯一请求 ID, 由服务端生成, 每次请求都会返回 (若请求因其他原因未能抵达服务端, 则该次请求不会获得RequestId)。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 使用Filters查看伸缩活动列表

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAutoScalingActivities
<公共请求参数>

{
  "Filters": [
```

```
{
  "Values": [
    "asa-o4v87ae9"
  ],
  "Name": "activity-id"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "ActivitySet": [
      {
        "Description": "Activity was launched in response to a difference between desired capacity and actual capacity, scale out 1 instance(s).",
        "AutoScalingGroupId": "asg-2umy3jbd",
        "RelatedInstanceSet": [
          {
            "InstanceId": "ins-q3ss14yo",
            "InstanceStatus": "SUCCESSFUL"
          }
        ],
        "LifecycleActionResultSet": [],
        "DetailedStatusMessageSet": [],
        "ActivityType": "SCALE_OUT",
        "ActivityId": "asa-o4v87ae9",
        "StartTime": "2018-11-20T08:33:56Z",
        "CreatedTime": "2018-11-20T08:33:56Z",
        "EndTime": "2018-11-20T08:34:52Z",
        "Cause": "Activity was launched in response to a difference between desired capacity and actual capacity.",
        "StatusMessageSimplified": "Success",
        "StatusMessage": "Success",
        "StatusCode": "SUCCESSFUL",
        "InvocationResultSet": []
      }
    ],
    "RequestId": "1082ab5d-c985-4d8c-bb9d-0d05e282b4a7"
  }
}
```

示例2 根据伸缩活动ID查询伸缩活动列表

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAutoScalingActivities
<公共请求参数>
```

```
{
  "ActivityIds": [
    "asa-o4v87ae9"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "ActivitySet": [
      {
        "Description": "Activity was launched in response to a difference between desired capacity and actual capacity, scale out 1 instance(s).",
        "AutoScalingGroupId": "asg-2umy3jbd",
        "RelatedInstanceSet": [
          {
            "InstanceId": "ins-q3ss14yo",
            "InstanceStatus": "SUCCESSFUL"
          }
        ],
        "LifecycleActionResultSet": [],
        "DetailedStatusMessageSet": [],
        "ActivityType": "SCALE_OUT",
        "ActivityId": "asa-o4v87ae9",
        "StartTime": "2018-11-20T08:33:56Z",
        "CreatedTime": "2018-11-20T08:33:56Z",
        "EndTime": "2018-11-20T08:34:52Z",
        "Cause": "Activity was launched in response to a difference between desired capacity and actual capacity.",
        "StatusMessageSimplified": "Success",
        "StatusMessage": "Success",
        "StatusCode": "SUCCESSFUL",
        "InvocationResultSet": []
      }
    ],
    "RequestId": "1082ab5d-c985-4d8c-bb9d-0d05e282b4a7"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue.Filter	无效的过滤器。
InvalidParameterValue.InvalidActivityId	伸缩活动ID无效。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidFilter	过滤条件无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
LimitExceeded.FilterValuesTooLong	特定过滤器的值过多。

查询伸缩组最新一次伸缩活动

最近更新时间：2025-05-16 01:09:35

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeAutoScalingGroupLastActivities）用于查询伸缩组的最新一次伸缩活动记录。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeAutoScalingGroupLastActivities。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupIds.N	是	Array of String	伸缩组ID列表。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：["asg-2umy3jbd"]
ExcludeCancelledActivity	否	Boolean	查询时排除取消类型活动。默认值为 false，表示不排除取消类型活动。 示例值：false

3. 输出参数

参数名称	类型	描述
ActivitySet	Array of Activity	符合条件的伸缩活动信息集合。说明：伸缩组伸缩活动不存在的则不返回，如传50个伸缩组ID，返回45条数据，说明其中有5个伸缩组伸缩活动不存在。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询伸缩组最新一次伸缩活动列表

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAutoScalingGroupLastActivities
<公共请求参数>
```

```
{
  "AutoScalingGroupIds": [
    "asg-2umy3jbd"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivitySet": [
      {
        "Description": "Activity was launched in response to a difference between desired capacity and actual capacity, scale out 1 instance(s).",
        "AutoScalingGroupId": "asg-2umy3jbd",
        "LifecycleActionResultSet": [],
        "RelatedInstanceSet": [
          {
            "InstanceId": "ins-q3ss14yo",
            "InstanceStatus": "SUCCESSFUL"
          }
        ],
        "DetailedStatusMessageSet": [],
        "ActivityType": "SCALE_OUT",
        "ActivityId": "asa-o4v87ae9",
        "StartTime": "2018-11-20T08:33:56Z",
        "CreatedTime": "2018-11-20T08:33:56Z",
        "EndTime": "2018-11-20T08:34:52Z",
        "Cause": "Activity was launched in response to a difference between desired capacity and actual capacity.",
        "StatusMessageSimplified": "Success",
        "StatusMessage": "Success",
        "StatusCode": "SUCCESSFUL",
        "InvocationResultSet": []
      }
    ],
    "RequestId": "1082ab5d-c985-4d8c-bb9d-0d05e282b4a7"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.NoResourcePermission	无资源权限。

删除伸缩组

最近更新时间：2025-05-16 01:09:35

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DeleteAutoScalingGroup）用于删除指定伸缩组，删除前提是伸缩组内无运行中（IN_SERVICE）状态的实例且当前未在执行伸缩活动。删除伸缩组后，创建失败（CREATION_FAILED）、中止失败（TERMINATION_FAILED）、解绑失败（DETACH_FAILED）等非运行中状态的实例不会被销毁。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteAutoScalingGroup。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-nvnlpbb8

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除指定伸缩组

删除伸缩组asg-nvnlpbb8

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteAutoScalingGroup
<公共请求参数>
```

```
{
```

```
"AutoScalingGroupId": "asg-nvnlpbb8"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "b12066f8-1a4d-4a08-a0e0-d4b8a7f88568"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallError	内部接口调用失败。
InternalServerError.CallMonitorError	Monitor接口调用失败。
InternalServerError.CallTagError	Tag 接口调用失败。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
ResourceInUse.ActivityInProgress	伸缩组正在执行伸缩活动。

错误码	描述
ResourceInUse.InstanceInGroup	伸缩组内尚有正常实例。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。

修改伸缩组

最近更新时间：2025-06-11 01:05:03

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ModifyAutoScalingGroup）用于修改伸缩组。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyAutoScalingGroup。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-gf5r6t7u
AutoScalingGroupName	否	String	伸缩组名称，在您账号中必须唯一。名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点，最大长度不能超55个字节。示例值：asgname
DefaultCooldown	否	Integer	默认冷却时间，单位秒，取值范围 [0,3600]，默认值为300。示例值：300
DesiredCapacity	否	Integer	期望实例数，取值范围 [0,2000]。需满足最大值大于等于期望值，期望值大于等于最小值。示例值：1
LaunchConfigurationId	否	String	启动配置ID。可以通过如下方式获取可用的启动配置ID： <ul style="list-style-type: none">通过登录 控制台 查询启动配置ID。通过调用接口 DescribeLaunchConfigurations，取返回信息中的 LaunchConfigurationId 获取启动配置ID。 示例值：asc-gf5r6t7u
MaxSize	否	Integer	最大实例数，取值范围为 [0,2000]。需满足最大值大于等于期望值，期望值大于等于最小值。示例值：1

参数名称	必选	类型	描述
MinSize	否	Integer	最小实例数，取值范围为 [0,2000]。需满足最大值大于等于期望值，期望值大于等于最小值。 示例值：1
ProjectId	否	Integer	项目ID。该参数可以通过调用 DescribeProject 的返回值中的 ProjectId 字段来获取。默认值为 0，表示使用默认项目。 示例值：1
SubnetIds.N	否	Array of String	子网ID列表。有效的私有网络子网ID可通过登录 控制台 查询；也可以调用接口 DescribeSubnets ，从接口返回中的 SubnetId 字段获取。 示例值：["subnet-gf5r6t7u"]
TerminationPolicies.N	否	Array of String	销毁策略，目前长度上限为1。取值包括 OLDEST_INSTANCE 和 NEWEST_INSTANCE。 <ul style="list-style-type: none"> OLDEST_INSTANCE 优先销毁伸缩组中最旧的实例。 NEWEST_INSTANCE，优先销毁伸缩组中最新的实例。 示例值：["OLDEST_INSTANCE"]
VpcId	否	String	私有网络ID。修改私有网络时，需将 SubnetIds 参数同步修改为该私有网络的子网。有效的 VpcId 可通过登录 控制台 查询；也可以调用接口 DescribeVpc ，从接口返回中的 VpcId 字段获取。 示例值：vpc-gf5r6t7u
Zones.N	否	Array of String	可用区列表 示例值：["ap-beijing-1"]
RetryPolicy	否	String	重试策略，取值包括 IMMEDIATE_RETRY、INCREMENTAL_INTERVALS、NO_RETRY，默认取值为 IMMEDIATE_RETRY。部分成功的伸缩活动判定为一次失败活动。 <ul style="list-style-type: none"> IMMEDIATE_RETRY，立即重试，在较短时间内快速重试，连续失败超过一定次数（5次）后不再重试。 INCREMENTAL_INTERVALS，间隔递增重试，随着连续失败次数的增加，重试间隔逐渐增大，重试间隔从秒级到1天不等。 NO_RETRY，不进行重试，直到再次收到用户调用或者告警信息后才会重试。 示例值：IMMEDIATE_RETRY
ZonesCheckPolicy	否	String	可用区校验策略，取值包括 ALL 和 ANY，默认取值为 ANY。在伸缩组实际变更资源相关字段时（启动配置、可用区、子网）发挥作用。 <ul style="list-style-type: none"> ALL，所有可用区（Zone）或子网（SubnetId）都可用则通过校验，否则校验报错。 ANY，存在任何一个可用区（Zone）或子网（SubnetId）可用则通过校验，否则校验报错。 可用区或子网不可用的常见原因包括该可用区CVM实例类型售罄、该可用区CBS云盘售罄、该可用区配额不足、该子网IP不足等。 如果 Zones/SubnetIds 中可用区或者子网不存在，则无论 ZonesCheckPolicy 采用何种取值，都会校验报错。 示例值：ALL

参数名称	必选	类型	描述
ServiceSettings	否	ServiceSettings	服务设置，包括云监控不健康替换等服务设置。 示例值：{"ReplaceLoadBalancerUnhealthy": "false", "ReplaceMonitorUnhealthy": "false"}
Ipv6AddressCount	否	Integer	实例具有IPv6地址数量的配置，取值包括0、1。默认值为0，表示实例不分配IPv6地址。需使用支持IPv6的私有网络，需在子网中开启IPv6 CIDR，其他使用限制可参考 IPv6使用限制 。 示例值：1
MultiZoneSubnetPolicy	否	String	多可用区/子网策略，取值包括 PRIORITY 和 EQUALITY，默认为 PRIORITY。 <ul style="list-style-type: none"> PRIORITY，按照可用区/子网列表的顺序，作为优先级来尝试创建实例，如果优先级最高的可用区/子网可以创建成功，则总在该可用区/子网创建。 EQUALITY：扩容出的实例会打散到多个可用区/子网，保证扩容后的各个可用区/子网实例数相对均衡。 <p>与本策略相关的注意点：</p> <ul style="list-style-type: none"> 当伸缩组为基础网络时，本策略适用于多可用区；当伸缩组为VPC网络时，本策略适用于多子网，此时不再考虑可用区因素，例如四个子网ABCD，其中ABC处于可用区1，D处于可用区2，此时考虑子网ABCD进行排序，而不考虑可用区1、2。 本策略适用于多可用区/子网，不适用于启动配置的多机型。多机型按照优先级策略进行选择。 按照 PRIORITY 策略创建实例时，先保证多机型的策略，后保证多可用区/子网的策略。例如多机型A、B，多子网1、2、3，会按照A1、A2、A3、B1、B2、B3 进行尝试，如果A1售罄，会尝试A2（而非B1）。 <p>示例值：PRIORITY</p>
HealthCheckType	否	String	伸缩组实例健康检查类型，取值如下： <ul style="list-style-type: none"> CVM：根据实例网络状态判断实例是否处于不健康状态，不健康的网络状态即发生实例 PING 不可达事件，详细判断标准可参考实例健康检查 CLB：根据 CLB 的健康检查状态判断实例是否处于不健康状态，CLB健康检查原理可参考健康检查 <p>示例值：CLB</p>
LoadBalancerHealthCheckGracePeriod	否	Integer	CLB健康检查宽限期。 示例值：0
InstanceAllocationPolicy	否	String	实例分配策略，取值包括 LAUNCH_CONFIGURATION 和 SPOT_MIXED。 <ul style="list-style-type: none"> LAUNCH_CONFIGURATION，代表传统的按照启动配置模式。 SPOT_MIXED，代表竞价混合模式。目前仅支持启动配置为按量计费模式时使用混合模式，混合模式下，伸缩组将根据设定扩容按量或竞价机型。使用混合模式时，关联的启动配置的计费类型不可被修改。 <p>示例值：LAUNCH_CONFIGURATION</p>

参数名称	必选	类型	描述
SpotMixedAllocationPolicy	否	SpotMixedAllocationPolicy	竞价混合模式下，各计费类型实例的分配策略。仅当 InstanceAllocationPolicy 取 SPOT_MIXED 时可用。 示例值: {"BaseCapacity": 1, "CompensateWithBaseInstance": true, "OnDemandPercentageAboveBaseCapacity": 50, "SpotAllocationStrategy": "COST_OPTIMIZED"}
CapacityRebalance	否	Boolean	容量重平衡功能，仅对伸缩组内的竞价实例有效。取值范围： <ul style="list-style-type: none"> TRUE，开启该功能，当伸缩组内的竞价实例即将被竞价实例服务自动回收前，AS 主动发起竞价实例销毁流程，如果有配置过缩容 hook，则销毁前 hook 会生效。销毁流程启动后，AS 会异步开启一个扩容活动，用于补齐期望实例数。 FALSE，不开启该功能，则 AS 等待竞价实例被销毁后才会去扩容补齐伸缩组期望实例数。 示例值: false
InstanceNameIndexSettings	否	InstanceNameIndexSettings	实例名称序号相关设置。开启后为伸缩组内自动创建的实例名称添加递增的数字序号。 示例值: {"BeginIndex": 0, "Enabled": true}

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改伸缩组VPC子网信息

修改指定伸缩组的VpcId和SubnetIds参数

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyAutoScalingGroup
<公共请求参数>

{
  "AutoScalingGroupId": "asg-ka0s0q80",
  "SubnetIds": [
    "subnet-b0vxjhot",
    "subnet-3tmer137"
  ],
  "VpcId": "vpc-hy436tmc"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c503ddc6-496c-44c9-8cec-e9f1c3f9c11c"
  }
}
```

示例2 修改伸缩组期望实例、最大实例数、最小实例数

修改指定伸缩组的期望实例数为3，最大实例数为10，最小实例数为1

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyAutoScalingGroup
<公共请求参数>

{
  "AutoScalingGroupId": "asg-ka0s0q80",
  "MinSize": 1,
  "MaxSize": 10,
  "DesiredCapacity": 3
}
```

输出示例

```
{
  "Response": {
    "RequestId": "b41d8d30-21d4-412c-b7f3-53041879968c"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalError.CallVpcError	VPC接口调用失败。
InternalError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameterValue.AssertDesiredCapacityFailed	声明的原有期望实例数和实际原有期望实例数不一致。
InvalidParameterValue.BaseCapacityTooLarge	指定的基础容量过大，需小于等于最大实例数。
InvalidParameterValue.CvmError	CVM参数校验异常。
InvalidParameterValue.DuplicatedSubnet	指定的子网重复。
InvalidParameterValue.GroupNameDuplicated	伸缩组名称重复。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidLaunchConfigurationId	启动配置ID无效。
InvalidParameterValue.InvalidSubnetId	子网ID无效。
InvalidParameterValue.LaunchConfigurationNotFound	找不到指定启动配置。
InvalidParameterValue.LbProjectInconsistent	负载均衡器项目不一致。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.OnlyVpc	账号仅支持VPC网络。
InvalidParameterValue.ProjectIdNotFound	项目ID不存在。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.Size	伸缩组最大数量、最小数量、期望实例数取值不合法。
InvalidParameterValue.SubnetIds	子网信息不合法。
InvalidParameterValue.SubnetNotInDedicatedCluster	子网不在指定本地专用集群中。
InvalidParameterValue.TooLong	取值过多。
LimitExceeded	超过配额限制。
LimitExceeded.MaxSizeLimitExceeded	最大实例数大于限制。
LimitExceeded.MinSizeLimitExceeded	最小实例数低于限制。
MissingParameter	缺少参数错误。
MissingParameter.InScenario	在特定场景下缺少参数。

错误码	描述
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceUnavailable.ForbiddenModifyVpc	绑定负载均衡器的伸缩组禁止修改Vpc。
ResourceUnavailable.LaunchConfigurationStatusAbnormal	启动配置状态异常。
ResourceUnavailable.ProjectInconsistent	项目不一致。

查询实例

最近更新时间：2025-05-16 01:09:34

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeAutoScalingInstances）用于查询弹性伸缩关联实例的信息。

- 可以根据实例ID、伸缩组ID等信息来查询实例的详细信息。过滤信息详细请见过滤器Filter。
- 如果参数为空，返回当前用户一定数量（Limit所指定的数量，默认为20）的实例。

默认接口请求频率限制：60次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeAutoScalingInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceIds.N	否	Array of String	待查询云服务器（CVM）的实例ID列表，列表长度上限为100，不支持同时指定 InstanceIds 和 Filters。 可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">通过登录控制台查询实例ID。通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-1fswxz1m"]
Filters.N	否	Array of Filter	过滤条件。 <ul style="list-style-type: none">instance-id - String - 是否必填：否 - （过滤条件）按照实例ID过滤。可通过登录控制台或调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。auto-scaling-group-id - String - 是否必填：否 - （过滤条件）按照伸缩组ID过滤。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 每次请求的Filters的上限为10，Filter.Values的上限为5。参数不支持同时指定InstanceIds和Filters。 示例值：[{"Name": "auto-scaling-group-id", "Values": ["asg-ewdpp04x"]}]
Offset	否	Integer	偏移量，默认为0。关于Offset的更进一步介绍请参考 API 简介 中的相关小节。 示例值：0
Limit	否	Integer	返回数量，默认为20，最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值：20

3. 输出参数

参数名称	类型	描述
AutoScalingInstanceSet	Array of Instance	实例详细信息列表。
TotalCount	Integer	符合条件的实例数量。 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询实例

根据指定实例 ID 或过滤条件查询实例的相关信息。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAutoScalingInstances
<公共请求参数>

{
  "InstanceIds": [
    "ins-1fswxz1m"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "AutoScalingInstanceSet": [
      {
        "ProtectedFromScaleIn": false,
        "Zone": "ap-guangzhou-3",
        "LaunchConfigurationId": "asc-5fzsm72a",
        "InstanceId": "ins-1fswxz1m",
        "VersionNumber": 1,
        "DisasterRecoverGroupIds": [],
        "AddTime": "2018-08-21T12:05:12Z",
        "CreationType": "AUTO_CREATION",
        "AutoScalingGroupName": "asgname",
        "AutoScalingGroupId": "asg-4o61gsxi",
        "HealthStatus": "HEALTHY",
        "LifecycleState": "IN_SERVICE",
        "LaunchConfigurationName": "系列2本地盘",
        "InstanceType": "S2.SMALL2",
        "WarmupStatus": "IN_WARMUP"
      }
    ],
  }
```

```
"RequestId": "2ae3e836-d47a-431c-b54b-4e1c2f419e5b"
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue.Filter	无效的过滤器。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidFilter	过滤条件无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
LimitExceeded.FilterValuesTooLong	特定过滤器的值过多。

查询伸缩组

最近更新时间：2025-05-16 01:09:34

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeAutoScalingGroups）用于查询伸缩组信息。

- 可以根据伸缩组ID、伸缩组名称或者启动配置ID等信息来查询伸缩组的详细信息。过滤信息详细请见过滤器Filter。
- 如果参数为空，返回当前用户一定数量（Limit所指定的数量，默认为20）的伸缩组。

默认接口请求频率限制：60次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeAutoScalingGroups。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupIds.N	否	Array of String	按照一个或者多个伸缩组ID查询。伸缩组ID形如：asg-nkdwoui0。每次请求的上限为100。参数不支持同时指定AutoScalingGroupIds和Filters。 示例值：["asg-nkdwoui0"]
Filters.N	否	Array of Filter	过滤条件。 <ul style="list-style-type: none"> auto-scaling-group-id - String - 是否必填：否 - （过滤条件）按照伸缩组ID过滤。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups，取返回信息中的AutoScalingGroupId 获取伸缩组ID。 auto-scaling-group-name - String - 是否必填：否 - （过滤条件）按照伸缩组名称过滤。 vague-auto-scaling-group-name - String - 是否必填：否 - （过滤条件）按照伸缩组名称模糊搜索。 launch-configuration-id - String - 是否必填：否 - （过滤条件）按照启动配置ID过滤。可通过登录 控制台 或调用接口 DescribeLaunchConfigurations，取返回信息中的LaunchConfigurationId 获取启动配置ID。 tag-key - String - 是否必填：否 - （过滤条件）按照标签键进行过滤。可通过调用接口 GetTags，取返回信息中的 TagKey 获取标签键。 tag-value - String - 是否必填：否 - （过滤条件）按照标签值进行过滤。可通过调用接口 GetTags，取返回信息中的 TagValue 获取标签值。 tag:tag-key - String - 是否必填：否 - （过滤条件）按照标签键值对进行过滤。tag-key 使用具体的标签键进行替换，请参考示例2。可通过调用接口 GetTags，取返回信息中的 TagKey 获取标签键。 每次请求的Filters的上限为10，Filter.Values的上限为5。参数不支持同时指定AutoScalingGroupIds和Filters。 示例值：[{"Name": "auto-scaling-group-name", "Values": ["production-as"]}]]
Limit	否	Integer	返回数量，默认为20，最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值：10

参数名称	必选	类型	描述
Offset	否	Integer	偏移量，默认为0。关于Offset的更进一步介绍请参考 API 简介中的相关小节。 示例值：0

3. 输出参数

参数名称	类型	描述
AutoScalingGroupSet	Array of AutoScalingGroup	伸缩组详细信息列表。
TotalCount	Integer	符合条件的伸缩组数量。 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询伸缩组

指定伸缩组ID查询伸缩组

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAutoScalingGroups
<公共请求参数>

{
  "AutoScalingGroupIds": [
    "asg-pts6qcgp"
  ]
}
```

输出示例

```
{
  "Response": {
    "AutoScalingGroupSet": [
      {
        "InActivityStatus": "NOT_IN_ACTIVITY",
        "LoadBalancerIdSet": [],
        "InstanceNameIndexSettings": {
          "BeginIndex": 0,
          "Enabled": true
        },
        "RetryPolicy": "IMMEDIATE_RETRY",
        "InServiceInstanceCount": 0,
        "CreatedTime": "2022-04-21T03:21:14Z",
        "SpotMixedAllocationPolicy": {
          "CompensateWithBaseInstance": null,
```

```
"SpotAllocationStrategy": null,
"OnDemandPercentageAboveBaseCapacity": null,
"BaseCapacity": null
},
"VpcId": "vpc-lceuvai4",
"InstanceAllocationPolicy": "LAUNCH_CONFIGURATION",
"Tags": [],
"LaunchConfigurationId": "asc-mo1woym9",
"MaxSize": 1,
"MultiZoneSubnetPolicy": "PRIORITY",
"SubnetIdSet": [
  "subnet-6qqolfi7"
],
"HealthCheckType": "CLB",
"LoadBalancerHealthCheckGracePeriod": 0,
"ForwardLoadBalancerSet": [
  {
    "TargetAttributes": [
      {
        "Port": 8080,
        "Weight": 10
      }
    ],
    "Region": "ap-shanghai",
    "LocationId": "loc-5dovrfov",
    "ListenerId": "lbl-i4p05pmv",
    "LoadBalancerId": "lb-pbx8nq2p"
  }
],
"ProjectId": 0,
"AutoScalingGroupName": "product-asg-001",
"MinSize": 0,
"ServiceSettings": {
  "ReplaceMonitorUnhealthy": false,
  "ReplaceLoadBalancerUnhealthy": false,
  "ScalingMode": "CLASSIC_SCALING",
  "DesiredCapacitySyncWithMaxMinSize": false,
  "ReplaceMode": "RECREATE"
},
"LaunchConfigurationName": "product-asc-001",
"CapacityRebalance": false,
"TerminationPolicySet": [
  "OLDEST_INSTANCE"
],
"AutoScalingGroupStatus": "NORMAL",
"InstanceCount": 0,
"DesiredCapacity": 0,
"AutoScalingGroupId": "asg-pts6qcgp",
"Ipv6AddressCount": 0,
"DefaultCooldown": 300,
"EnabledStatus": "ENABLED",
"ZoneSet": []
}
```



```
],
"TotalCount": 1,
"RequestId": "53a76c96-a88c-4972-8488-66d6c15a080f"
}
}
```

示例2 查询绑定了标签的伸缩组

查询绑定了标签键值对（city:shenzhen）的伸缩组。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAutoScalingGroups
<公共请求参数>
```

```
{
  "Limit": 1,
  "Filters": [
    {
      "Values": [
        "shenzhen"
      ],
      "Name": "tag:city"
    }
  ],
  "Offset": 0
}
```

输出示例

```
{
  "Response": {
    "AutoScalingGroupSet": [
      {
        "InActivityStatus": "NOT_IN_ACTIVITY",
        "LoadBalancerIdSet": [],
        "InstanceNameIndexSettings": {
          "BeginIndex": 0,
          "Enabled": true
        },
        "RetryPolicy": "IMMEDIATE_RETRY",
        "InServiceInstanceCount": 0,
        "CreatedTime": "2019-10-29T02:21:26Z",
        "SpotMixedAllocationPolicy": {
          "CompensateWithBaseInstance": null,
          "SpotAllocationStrategy": null,
          "OnDemandPercentageAboveBaseCapacity": null,
          "BaseCapacity": null
        },
        "VpcId": "vpc-qmjqjnk",
```

```
"InstanceAllocationPolicy": "LAUNCH_CONFIGURATION",
"Tags": [
{
"Key": "city",
"ResourceType": "auto-scaling-group",
"Value": "shenzhen"
}
],
"LaunchConfigurationId": "asc-3d9e2zfx",
"MaxSize": 10,
"MultiZoneSubnetPolicy": "PRIORITY",
"SubnetIdSet": [
"subnet-3cpb9yfp",
"subnet-c98udmmr",
"subnet-1xsr551x",
"subnet-o3ibshdr",
"subnet-6c7q2jhz"
],
"HealthCheckType": "CVM",
"LoadBalancerHealthCheckGracePeriod": 0,
"ForwardLoadBalancerSet": [
{
"TargetAttributes": [
{
"Port": 8080,
"Weight": 10
}
],
"Region": "ap-shanghai",
"LocationId": "loc-y6t5rew2",
"ListenerId": "lbl-aiwdu9bd",
"LoadBalancerId": "lb-k264wzwj"
},
{
"TargetAttributes": [
{
"Port": 80,
"Weight": 10
}
],
"Region": "ap-shanghai",
"LocationId": "loc-qmxmx085",
"ListenerId": "lbl-ldjbrn65",
"LoadBalancerId": "lb-k264wzwj"
}
],
"ProjectId": 0,
"AutoScalingGroupName": "sz-asg",
"MinSize": 0,
"ServiceSettings": {
"ReplaceMonitorUnhealthy": false,
"ReplaceLoadBalancerUnhealthy": false,
"ReplaceMode": "RECREATE",
```

```
"DesiredCapacitySyncWithMaxMinSize": false,
"ScalingMode": "CLASSIC_SCALING"
},
"LaunchConfigurationName": "sz-asc",
"CapacityRebalance": false,
"TerminationPolicySet": [
  "OLDEST_INSTANCE"
],
"AutoScalingGroupStatus": "NORMAL",
"InstanceCount": 0,
"DesiredCapacity": 0,
"AutoScalingGroupId": "asg-h71x7a3f",
"Ipv6AddressCount": 0,
"DefaultCooldown": 300,
"EnabledStatus": "ENABLED",
"ZoneSet": []
}
],
"TotalCount": 1,
"RequestId": "53a76c96-a88c-4972-8488-66d6c15a080f"
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。

错误码	描述
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterConflict	指定的两个参数冲突，不能同时存在。
InvalidParameterValue.Filter	无效的过滤器。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidFilter	过滤条件无效。
InvalidParameterValue.InvalidLaunchConfigurationId	启动配置ID无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.TooLong	取值过多。
InvalidPermission	账户不支持该操作。
LimitExceeded.FilterValuesTooLong	特定过滤器的值过多。
UnsupportedOperation	操作不支持。

创建伸缩组

最近更新时间：2025-06-11 01:05:03

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（CreateAutoScalingGroup）用于创建伸缩组

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateAutoScalingGroup。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupName	是	String	伸缩组名称，在您账号中必须唯一。名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点，最大长度不能超55个字节。 示例值：asgname
LaunchConfigurationId	是	String	启动配置ID。可通过登录 控制台 或调用接口 DescribeLaunchConfigurations ，取返回信息中的 LaunchConfigurationId 获取启动配置ID。 示例值：asc-aaaabbbb
MaxSize	是	Integer	最大实例数，取值范围为 [0,2000]，同时需满足 MaxSize >= DesiredCapacity >= MinSize。 示例值：1
MinSize	是	Integer	最小实例数，取值范围为 [0,2000]，同时需满足 MaxSize >= DesiredCapacity >= MinSize。 示例值：1
VpcId	是	String	私有网络ID。有效的VpcId可通过登录 控制台 查询；也可以调用接口 DescribeVpc ，从接口返回中的VpcId 字段获取。 示例值：vpc-uhygtf56
DefaultCooldown	否	Integer	默认冷却时间，单位秒，默认值为300。取值范围为 [0,3600]。 示例值：300
DesiredCapacity	否	Integer	期望实例数，取值范围 [0,2000]，默认值等于当前 MinSize，同时需满足 MaxSize >= DesiredCapacity >= MinSize。 示例值：1

参数名称	必选	类型	描述
LoadBalancerIds.N	否	Array of String	传统负载均衡器ID列表，目前长度上限为20，LoadBalancerIds 和 ForwardLoadBalancers 二者同时最多只能指定一个。可以通过 DescribeLoadBalancers 接口获取。 示例值: ["lb-uhygtf56"]
ProjectId	否	Integer	伸缩组内实例所属项目ID。默认值为0，表示使用默认项目。该参数可以通过调用 DescribeProject 的返回中的 projectId 字段来获取。 示例值: 1
ForwardLoadBalancers.N	否	Array of ForwardLoadBalancer	负载均衡器列表，目前长度上限为100，LoadBalancerIds 和 ForwardLoadBalancers 二者同时最多只能指定一个 示例值: [{"LoadBalancerId": "lb-af8li2kn", "ListenerId": "lbi-akw1lgo7", "TargetAttributes": [{"Port": 80, "Weight": 10}]}]
SubnetIds.N	否	Array of String	子网ID列表，VPC场景下必须指定子网。多个子网以填写顺序为优先级，依次进行尝试，直至可以成功创建实例。有效的私有网络子网ID可通过登录 控制台 查询；也可以调用接口 DescribeSubnets ，从接口返回中的 SubnetId 字段获取。 示例值: ["subnet-uhygtf56"]
TerminationPolicies.N	否	Array of String	销毁策略，目前长度上限为1。取值包括 OLDEST_INSTANCE 和 NEWEST_INSTANCE，默认取值为 OLDEST_INSTANCE。 <ul style="list-style-type: none"> OLDEST_INSTANCE 优先销毁伸缩组中最旧的实例。 NEWEST_INSTANCE，优先销毁伸缩组中最新的实例。 示例值: OLDEST_INSTANCE
Zones.N	否	Array of String	可用区列表，基础网络场景下必须指定可用区。多个可用区以填写顺序为优先级，依次进行尝试，直至可以成功创建实例。 示例值: ["ap-beijing-1"]
RetryPolicy	否	String	重试策略，取值包括 IMMEDIATE_RETRY、INCREMENTAL_INTERVALS、NO_RETRY，默认取值为 IMMEDIATE_RETRY。部分成功的伸缩活动判定为一次失败活动。 <ul style="list-style-type: none"> IMMEDIATE_RETRY，立即重试，在较短时间内快速重试，连续失败超过一定次数（5次）后不再重试。 INCREMENTAL_INTERVALS，间隔递增重试，随着连续失败次数的增加，重试间隔逐渐增大。前10次重试间隔与立即重试模式相同，后续逐步递增至10分钟、30分钟、60分钟、一天。 NO_RETRY，不进行重试，直到再次收到用户调用或者告警信息后才会重试。 示例值: IMMEDIATE_RETRY
ZonesCheckPolicy	否	String	可用区校验策略，取值包括 ALL 和 ANY，默认取值为 ANY。 <ul style="list-style-type: none"> ALL，所有可用区（Zone）或子网（SubnetId）都可用则通过校验，否则校验报错。

参数名称	必选	类型	描述
			<ul style="list-style-type: none"> ANY, 存在任何一个可用区 (Zone) 或子网 (SubnetId) 可用则通过校验, 否则校验报错。 <p>可用区或子网不可用的常见原因包括该可用区CVM实例类型售罄、该可用区CBS云盘售罄、该可用区配额不足、该子网IP不足等。</p> <p>如果 Zones/SubnetIds 中可用区或者子网不存在, 则无论 ZonesCheckPolicy 采用何种取值, 都会校验报错。</p> <p>示例值: ALL</p>
Tags.N	否	Array of Tag	<p>标签描述列表。通过指定该参数可以支持绑定标签到伸缩组。同时绑定标签到相应的资源实例。每个伸缩组最多支持30个标签。可通过调用接口 GetTags, 根据回参获取已有的标签键值对信息。</p> <p>示例值: [{"Key": "Service_B", "Value": "pss3", "ResourceType": "auto-scaling-group"}]</p>
ServiceSettings	否	ServiceSettings	<p>服务设置, 包括云监控不健康替换等服务设置。</p> <p>示例值: {"ScalingMode": "CLASSIC_SCALING"}</p>
Ipv6AddressCount	否	Integer	<p>实例具有IPv6地址数量的配置, 取值包括0、1。默认值为0, 表示实例不分配IPv6地址。需使用支持IPv6的私有网络, 需在子网中开启IPv6 CIDR, 其他使用限制可参考 IPv6使用限制。</p> <p>示例值: 1</p>
MultiZoneSubnetPolicy	否	String	<p>多可用区/子网策略, 取值包括 PRIORITY 和 EQUALITY, 默认为 PRIORITY。</p> <ul style="list-style-type: none"> PRIORITY, 按照可用区/子网列表的顺序, 作为优先级来尝试创建实例, 如果优先级最高的可用区/子网可以创建成功, 则总在该可用区/子网创建。 EQUALITY: 扩容出的实例会打散到多个可用区/子网, 保证扩容后的各个可用区/子网实例数相对均衡。 <p>与本策略相关的注意点:</p> <ul style="list-style-type: none"> 当伸缩组为基础网络时, 本策略适用于多可用区; 当伸缩组为VPC网络时, 本策略适用于多子网, 此时不再考虑可用区因素, 例如四个子网ABCD, 其中ABC处于可用区1, D处于可用区2, 此时考虑子网ABCD进行排序, 而不考虑可用区1、2。 本策略适用于多可用区/子网, 不适用于启动配置的多机型。多机型按照优先级策略进行选择。 按照 PRIORITY 策略创建实例时, 先保证多机型的策略, 后保证多可用区/子网的策略。例如多机型A、B, 多子网1、2、3, 会按照A1、A2、A3、B1、B2、B3 进行尝试, 如果A1售罄, 会尝试A2 (而非B1)。 <p>示例值: PRIORITY</p>
HealthCheckType	否	String	<p>伸缩组实例健康检查类型, 取值如下:</p> <ul style="list-style-type: none"> CVM: 根据实例网络状态判断实例是否处于不健康状态, 不健康的网络状态即发生实例 PING 不可达事件, 详细判断标准可参考 实例健康检查 CLB: 根据 CLB 的健康检查状态判断实例是否处于不健康状态, CLB健康检查原理可参考 健康检查 <p>如果选择了CLB类型, 伸缩组将同时检查实例网络状态与CLB健康检查状态, 如果出现实例网络状态不健康,</p>

参数名称	必选	类型	描述
			实例将被标记为 UNHEALTHY 状态；如果出现 CLB 健康检查状态异常，实例将被标记为 CLB_UNHEALTHY 状态，如果两个异常状态同时出现，实例HealthStatus字段将返回 UNHEALTHY CLB_UNHEALTHY。默认值：CLB 示例值：CLB
LoadBalancerHealthCheckGracePeriod	否	Integer	CLB健康检查宽限期，当扩容的实例进入IN_SERVICE后，在宽限期时间范围内将不会被标记为不健康 CLB_UNHEALTHY。 默认值：0。取值范围[0, 7200]，单位：秒。 示例值：0
InstanceAllocationPolicy	否	String	实例分配策略，取值包括 LAUNCH_CONFIGURATION 和 SPOT_MIXED，默认取 LAUNCH_CONFIGURATION。 • LAUNCH_CONFIGURATION，代表传统的按照启动配置模式。 • SPOT_MIXED，代表竞价混合模式。目前仅支持启动配置为按量计费模式时使用混合模式，混合模式下，伸缩组将根据设定扩容按量或竞价机型。使用混合模式时，关联的启动配置的计费类型不可被修改。 示例值：LAUNCH_CONFIGURATION
SpotMixedAllocationPolicy	否	SpotMixedAllocationPolicy	竞价混合模式下，各计费类型实例的分配策略。仅当 InstanceAllocationPolicy 取 SPOT_MIXED 时可用。 示例值：{"BaseCapacity": 1, "CompensateWithBaseInstance": true, "OnDemandPercentageAboveBaseCapacity": 50, "SpotAllocationStrategy": "COST_OPTIMIZED"}
CapacityRebalance	否	Boolean	容量重平衡功能，仅对伸缩组内的竞价实例有效。取值范围： • TRUE，开启该功能，当伸缩组内的竞价实例即将被竞价实例服务自动回收前，AS 主动发起竞价实例销毁流程，如果有配置过缩容 hook，则销毁前 hook 会生效。销毁流程启动后，AS 会异步开启一个扩容活动，用于补齐期望实例数。 • FALSE，不开启该功能，则 AS 等待竞价实例被销毁后才会去扩容补齐伸缩组期望实例数。 默认取 FALSE。 示例值：false
InstanceNameIndexSettings	否	InstanceNameIndexSettings	实例名称序号相关设置。若不指定该参数，则默认不开启。开启后为伸缩组内自动创建的实例名称添加递增的数字序号。 示例值：{"BeginIndex": 0, "Enabled": true}

3. 输出参数

参数名称	类型	描述
AutoScalingGroupId	String	伸缩组ID 示例值：asg-y6tf4e3w

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建伸缩组

创建伸缩组，VPC网络，配置7层负载均衡器

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateAutoScalingGroup
<公共请求参数>

{
  "VpcId": "vpc-hy436tmc",
  "DesiredCapacity": 0,
  "LaunchConfigurationId": "asc-7vucy6ae",
  "ProjectId": 0,
  "SubnetIds": [
    "subnet-b0vxjhot",
    "subnet-3tmer137"
  ],
  "AutoScalingGroupName": "asg-vpc-7layer-lb",
  "DefaultCooldown": 300,
  "MinSize": 0,
  "MaxSize": 10,
  "TerminationPolicies": [
    "OLDEST_INSTANCE"
  ],
  "ForwardLoadBalancers": [
    {
      "TargetAttributes": [
        {
          "Port": 8080,
          "Weight": 10
        }
      ],
      "Region": "ap-guangzhou",
      "LocationId": "loc-13hmaev9",
      "ListenerId": "lbl-ncw704sn",
      "LoadBalancerId": "lb-23aejgcv"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AutoScalingGroupId": "asg-nkdwoui0",
    "RequestId": "a5d66fed-85b9-4f43-8243-597337ba896e"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallLbError	LB 接口调用失败。
InternalServerError.CallTagError	Tag 接口调用失败。
InternalServerError.CallTvpError	Tvp 接口调用失败。
InternalServerError.CallVpcError	VPC接口调用失败。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameterValue.BaseCapacityTooLarge	指定的基础容量过大，需小于等于最大实例数。
InvalidParameterValue.ClassicLb	在应当指定传统型负载均衡器的参数中，错误地指定了一个非传统型的负载均衡器。
InvalidParameterValue.CvmError	CVM参数校验异常。

错误码	描述
InvalidParameterValue.DuplicatedForwardLb	提供的应用型负载均衡器重复。
InvalidParameterValue.DuplicatedSubnet	指定的子网重复。
InvalidParameterValue.ForwardLb	在应当指定应用型负载均衡器的参数中，错误地指定了一个非应用型的负载均衡器。
InvalidParameterValue.GroupNameDuplicated	伸缩组名称重复。
InvalidParameterValue.InvalidClbRegion	为CLB指定的地域不合法。
InvalidParameterValue.InvalidLaunchConfigurationId	启动配置ID无效。
InvalidParameterValue.InvalidSubnetId	子网ID无效。
InvalidParameterValue.LaunchConfigurationNotFound	找不到指定启动配置。
InvalidParameterValue.LbProjectInconsistent	负载均衡器项目不一致。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.ListenerTargetTypeNotSupported	不支持目标组类型的监听器。
InvalidParameterValue.OnlyVpc	账号仅支持VPC网络。
InvalidParameterValue.ProjectIdNotFound	项目ID不存在。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.Size	伸缩组最大数量、最小数量、期望实例数取值不合法。
InvalidParameterValue.SubnetIds	子网信息不合法。
InvalidParameterValue.SubnetNotInDedicatedCluster	子网不在指定本地专用集群中。
InvalidParameterValue.TargetPortDuplicated	负载均衡器四层监听器的后端端口重复。
InvalidParameterValue.TooLong	取值过多。
InvalidParameterValue.ZoneMismatchRegion	指定的可用区与地域不匹配。
LimitExceeded	超过配额限制。
LimitExceeded.AutoScalingGroupLimitExceeded	伸缩组数量超过限制。
LimitExceeded.MaxSizeLimitExceeded	最大实例数大于限制。
LimitExceeded.MinSizeLimitExceeded	最小实例数低于限制。
MissingParameter.InScenario	在特定场景下缺少参数。
ResourceNotFound.ListenerNotFound	指定的Listener不存在。
ResourceNotFound.LoadBalancerNotFound	找不到指定负载均衡器。
ResourceNotFound.LocationNotFound	指定的Location不存在。
ResourceUnavailable.LaunchConfigurationStatusAbnormal	启动配置状态异常。
ResourceUnavailable.LbBackendRegionInconsistent	CLB实例的后端地域与AS服务所在地域不一致。
ResourceUnavailable.LbVpcInconsistent	负载均衡器VPC与伸缩组不一致。
ResourceUnavailable.ProjectInconsistent	项目不一致。
ResourceUnavailable.ZoneUnavailable	指定的可用区不可用。

指定数量扩容实例

最近更新时间：2025-04-25 01:08:50

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

为伸缩组指定数量扩容实例，返回扩容活动的 ActivityId。

- 伸缩组需要未处于活动中
- 伸缩组处于停用状态时，该接口也会生效，可参考[停用伸缩组](#)文档查看伸缩组停用状态的影响范围
- 接口会增加期望实例数，新的期望实例数需要小于等于最大实例数
- 扩容如果失败或者部分成功，最后期望实例数只会增加实际成功的实例数量
- 竞价混合模式中一次扩容可能触发多个伸缩活动，该接口仅返回第一个伸缩活动的 ActivityId

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ScaleOutInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。 示例值：asg-12yqet78
ScaleOutNumber	是	Integer	希望扩容的实例数量。 示例值：1

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID。 示例值：asa-n6w01f6m
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 伸缩组扩容一台实例

伸缩组asg-12yqet78扩容一台实例

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ScaleOutInstances
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12yqet78",
  "ScaleOutNumber": 1
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-k1q8oaz6",
    "RequestId": "6af368fd-35ff-4dcc-b302-35c378f2cccb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。

错误码	描述
InvalidParameterValue.Range	取值超出指定范围。
LimitExceeded.DesiredCapacityLimitExceeded	期望实例数超出限制。
ResourceInsufficient.AutoScalingGroupAboveMaxSize	超过伸缩组最大实例数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。

指定数量缩容实例

最近更新时间：2025-05-14 01:09:27

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

为伸缩组指定数量缩容实例，返回缩容活动的 ActivityId。

- 伸缩组需要未处于活动中
- 伸缩组处于停用状态时，该接口也会生效，可参考[停用伸缩组](#)文档查看伸缩组停用状态的影响范围
- 根据伸缩组的TerminationPolicies策略，选择被缩容的实例，可参考[缩容处理](#)
- 接口只会选择IN_SERVICE实例缩容，如果需要缩容其他状态实例，可以使用 [DetachInstances](#) 或 [RemoveInstances](#) 接口
- 接口会减少期望实例数，新的期望实例数需要大于等于最小实例数
- 缩容如果失败或者部分成功，最后期望实例数只会扣减实际缩容成功的实例数量

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ScaleInInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-12yqet78
ScaleInNumber	是	Integer	希望缩容的实例数量。该参数的静态取值范围是 [1,2000]，同时该参数不得大于期望数与最小值的差值。例如伸缩组期望数为 100，最小值为 20，此时可取值范围为 [1,80]。示例值：1

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID。 示例值：asa-k1q8oaz6
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 伸缩组缩容一台实例

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ScaleInInstances
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12yqet78",
  "ScaleInNumber": 1
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-n6w01f6m",
    "RequestId": "c0bb46ea-2b47-471c-9099-e20bf7a23078"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.Range	取值超出指定范围。
MissingParameter	缺少参数错误。
ResourceInsufficient.AutoScalingGroupBelowMinSize	少于伸缩组最小实例数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。

查询弹性伸缩配置建议

最近更新时间：2025-05-16 01:09:35

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

此接口用于查询伸缩组配置建议。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeAutoScalingAdvices。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupIds.N	是	Array of String	待查询的伸缩组列表，上限100。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：["asg-2pvrsyog", "asg-4wtj6aeq"]

3. 输出参数

参数名称	类型	描述
AutoScalingAdviceSet	Array of AutoScalingAdvice	伸缩组配置建议集合。 示例值：[{"AutoScalingGroupId": "asg-1c2n0mzp", "Advices": [], "Level": "NORMAL"}]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询伸缩组配置建议

查询伸缩组asg-2pvrsyog的配置建议

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAutoScalingAdvices
<公共请求参数>
```

```
{
```

```
"AutoScalingGroupIds": [
  "asg-2pvrsyog"
]
```

输出示例

```
{
  "Response": {
    "AutoScalingAdviceSet": [
      {
        "AutoScalingGroupId": "asg-2pvrsyog",
        "Level": "WARNING",
        "Advices": [
          {
            "Problem": "InvalidInstanceType",
            "Solution": "It is recommended to replace the invalid instance type.",
            "Detail": "Instance Type `S2.MEDIUM4`(`POSTPAID_BY_HOUR`) in `ap-guangzhou-1` is invalid.",
            "Level": "WARNING"
          },
          {
            "Problem": "InvalidInstanceType",
            "Solution": "It is recommended to replace the invalid instance type.",
            "Detail": "Instance Type `S2.MEDIUM4`(`POSTPAID_BY_HOUR`) in `ap-guangzhou-4` is invalid.",
            "Level": "WARNING"
          }
        ]
      }
    ],
    "RequestId": "5f7c48b7-222b-4f13-ae5e-978716b49f1c"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.LimitExceeded	取值超出限制。

伸缩组实例相关接口

从伸缩组中删除 CVM 实例

最近更新时间：2025-04-28 01:09:39

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（RemoveInstances）用于从伸缩组删除 CVM 实例。根据当前的产品逻辑，如果实例由弹性伸缩自动创建，则实例会被销毁；如果实例系创建后加入伸缩组的，则会从伸缩组中移除，保留实例。

- 如果删除指定实例后，伸缩组内处于IN_SERVICE状态的实例数量小于伸缩组最小值，接口将报错
- 如果伸缩组处于DISABLED状态，删除操作不校验IN_SERVICE实例数量和最小值的关系
- 对于伸缩组配置的 CLB，实例在离开伸缩组时，AS 会进行解挂载动作

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：RemoveInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">• 通过登录 控制台 查询伸缩组ID。• 通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-boz1qhnk
InstanceIds.N	是	Array of String	CVM实例ID列表。可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">• 通过登录控制台查询实例ID。• 通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-cri8d02t","ins-osckfnm7"]

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-dne04cxp
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 从伸缩组删除实例

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: RemoveInstances
<公共请求参数>

{
  "AutoScalingGroupId": "asg-boz1qhnk",
  "InstanceIds": [
    "ins-osckf7m7",
    "ins-cri8d02t"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-dne04cxp",
    "RequestId": "5b039ee6-e8ff-4605-bb24-b45337747431"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
ResourceInsufficient.AutoScalingGroupBelowMinSize	少于伸缩组最小实例数。
ResourceInsufficient.InServiceInstanceBelowMinSize	伸缩组内实例数低于最小实例数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.InstancesNotInAutoScalingGroup	目标实例不在伸缩组内。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。

从伸缩组移出 CVM 实例

最近更新时间：2025-04-28 01:09:39

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DetachInstances）用于从伸缩组移出 CVM 实例，本接口不会销毁实例。

- 如果移出指定实例后，伸缩组内处于IN_SERVICE状态的实例数量小于伸缩组最小值，接口将报错
- 如果伸缩组处于DISABLED状态，移出操作不校验IN_SERVICE实例数量和最小值的关系
- 对于伸缩组配置的 CLB，实例在离开伸缩组时，AS 会进行解挂载动作

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DetachInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-boz1qhnk
InstanceIds.N	是	Array of String	CVM实例ID列表。可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">通过登录控制台查询实例ID。通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-cri8d02t","ins-osckfnm7"]

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-bcfxy55
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 从伸缩组移出实例

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DetachInstances
<公共请求参数>

{
  "AutoScalingGroupId": "asg-boz1qhnk",
  "InstanceIds": [
    "ins-osckfnm7",
    "ins-cri8d02t"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-bcfxhy55",
    "RequestId": "5b039ee6-e8ff-4605-bb24-b45337747431"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
ResourceInsufficient.AutoScalingGroupBelowMinSize	少于伸缩组最小实例数。
ResourceInsufficient.InServiceInstanceBelowMinSize	伸缩组内实例数低于最小实例数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.InstancesNotInAutoScalingGroup	目标实例不在伸缩组内。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

添加 CVM 实例到伸缩组

最近更新时间：2025-04-28 01:09:39

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（AttachInstances）用于将 CVM 实例添加到伸缩组。

- 仅支持添加处于RUNNING（运行中）或STOPPED（已关机）状态的 CVM 实例
- 添加的 CVM 实例需要和伸缩组 VPC 网络一致

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：AttachInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-boz1qhmk
InstanceIds.N	是	Array of String	CVM实例ID列表。可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">通过登录控制台查询实例ID。通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-cri8d02t","ins-osckfnm7"]

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-q59zikez
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 添加实例到伸缩组中

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: AttachInstances
<公共请求参数>

{
  "AutoScalingGroupId": "asg-boz1qhnk",
  "InstanceIds": [
    "ins-osckfnm7",
    "ins-cri8d02t"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-q59zikez",
    "RequestId": "5b039ee6-e8ff-4605-bb24-b45337747431"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalServerError	内部错误。
InternalServerError.CalleeError	调用其他服务异常。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
ResourceInsufficient.AutoScalingGroupAboveMaxSize	超过伸缩组最大实例数。
ResourceInsufficient.InServiceInstanceAboveMaxSize	伸缩组内实例数超过最大实例数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.InstancesNotFound	指定的实例不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.CvmVpcInconsistent	实例和伸缩组Vpc不一致。
ResourceUnavailable.InstanceCannotAttach	实例无法添加至伸缩组。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。
ResourceUnavailable.InstancesAlreadyInAutoScalingGroup	实例已存在于伸缩组中。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

设置实例保护

最近更新时间：2025-04-28 01:09:38

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（SetInstancesProtection）用于设置实例保护。

实例设置保护之后，当发生不健康替换、报警策略、期望值变更等触发缩容时，将不对此实例缩容操作。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetInstancesProtection。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-2umy3jbd
InstanceIds.N	是	Array of String	实例ID。可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">通过登录 控制台 查询实例ID。通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-b2d33ywt"]
ProtectedFromScaleIn	是	Boolean	实例是否需要设置保护。 示例值：true

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 为实例设置保护

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: SetInstancesProtection
```

<公共请求参数>

```
{
  "AutoScalingGroupId": "asg-2umy3jbd",
  "ProtectedFromScaleIn": "true",
  "InstanceIds": [
    "ins-b2d33ywt"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "5b7168d9-5709-4d69-bd32-880a2f565e33"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。

错误码	描述
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.InstancesNotInAutoScalingGroup	目标实例不在伸缩组内。

关闭伸缩组内 CVM 实例

最近更新时间：2025-04-28 01:09:38

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（StopAutoScalingInstances）用于关闭伸缩组内 CVM 实例。

- 关机方式采用SOFT_FIRST方式，表示在正常关闭失败后进行强制关闭
- 关闭IN_SERVICE状态的实例，会减少期望实例数，期望实例数不可低于设置的最小值
- 使用STOP_CHARGING选项关机，待关机的实例需要满足[关机不收费条件](#)
- 本接口支持批量操作，每次请求关机实例的上限为100

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：StopAutoScalingInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">• 通过登录 控制台 查询伸缩组ID。• 通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-boz1qhnk
InstanceIds.N	是	Array of String	待关闭的CVM实例ID列表。可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">• 通过登录控制台查询实例ID。• 通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-cri8d02t","ins-osckfnm7"]
StoppedMode	否	String	关闭的实例是否收费，取值为： KEEP_CHARGING：关机继续收费 STOP_CHARGING：关机停止收费 默认为 KEEP_CHARGING 示例值：STOP_CHARGING

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-m1ebiskl

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 按关机不收费模式关闭伸缩组内实例

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: StopAutoScalingInstances
<公共请求参数>

{
  "AutoScalingGroupId": "asg-boz1qhnk",
  "StoppedMode": "STOP_CHARGING",
  "InstanceIds": [
    "ins-osckfnm7",
    "ins-cri8d02t"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-m1ebisk1",
    "RequestId": "f3e2873c-af7c-43ee-8aa7-53565d4181c2"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
CallCvmError	CVM接口调用失败。
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalServerError.CalleeError	调用其他服务异常。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
ResourceInsufficient.AutoScalingGroupBelowMinSize	少于伸缩组最小实例数。
ResourceInsufficient.InServiceInstanceBelowMinSize	伸缩组内实例数低于最小实例数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.InstancesNotInAutoScalingGroup	目标实例不在伸缩组内。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。
ResourceUnavailable.InstanceNotSupportStopCharging	实例不支持关机不收费。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

开启伸缩组内 CVM 实例

最近更新时间：2025-04-28 01:09:38

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（StartAutoScalingInstances）用于开启伸缩组内 CVM 实例。

- 开机成功，实例转为IN_SERVICE状态后，会增加期望实例数，期望实例数不可超过设置的最大值
- 本接口支持批量操作，每次请求开机实例的上限为100

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：StartAutoScalingInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">• 通过登录 控制台 查询伸缩组ID。• 通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-boz1qhmk
InstanceIds.N	是	Array of String	待开启的CVM实例ID列表。可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">• 通过登录控制台查询实例ID。• 通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-cri8d02t","ins-osckfnm7"]

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-a023dwdj
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 开启 CVM 实例

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: StartAutoScalingInstances
<公共请求参数>

{
  "AutoScalingGroupId": "asg-boz1qhnk",
  "InstanceIds": [
    "ins-osckfnm7",
    "ins-cri8d02t"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-a023dwdj",
    "RequestId": "28cf9089-2b76-4934-9d1b-b2694c679ff0"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
ResourceInsufficient.AutoScalingGroupAboveMaxSize	超过伸缩组最大实例数。
ResourceInsufficient.InServiceInstanceAboveMaxSize	伸缩组内实例数超过最大实例数。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.InstancesNotInAutoScalingGroup	目标实例不在伸缩组内。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

实例退出备用中状态

最近更新时间：2025-04-28 01:09:39

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

伸缩组内实例退出备用中状态。

- 退出备用中状态后，实例会进入运行中状态，CLB 权重值恢复为预设值
- 调用弹性伸缩开关机接口会使得备用中状态发生变化，而云服务器开关机接口不会影响
- 实例退出备用中状态后，伸缩组会上调期望实例数，新期望数不能大于最大值

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ExitStandby。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组 ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-boz1qhmk
InstanceIds.N	是	Array of String	备用中状态 CVM 实例列表。可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">通过登录控制台查询实例ID。通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-cri8d02t","ins-osckfnm7"]

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID。 示例值：asa-q59zikez
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 伸缩组内实例退出备用中状态

伸缩组 asg-boz1qhmk 中实例 ins-osckfnm7, ins-cri8d02t 退出备用中状态

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ExitStandby
<公共请求参数>

{
  "AutoScalingGroupId": "asg-boz1qhmk",
  "InstanceIds": [
    "ins-osckfnm7",
    "ins-cri8d02t"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-q59zikez",
    "RequestId": "5b039ee6-e8ff-4605-bb24-b45337747431"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalServerError	内部错误。
InternalServerError.CalleeError	调用其他服务异常。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.InstancesNotFound	指定的实例不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

实例进入备用中状态

最近更新时间：2025-04-28 01:09:39

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

伸缩组内实例进入备用中状态。

- 备用中状态实例的 CLB 权重值为 0，不会被自动缩容、不健康替换、实例刷新操作选中
- 调用弹性伸缩开关机接口会使得备用中状态发生变化，而云服务器开关机接口不会影响
- 实例进入备用中状态后，伸缩组会尝试下调期望实例数，新时期数不会小于最小值

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：EnterStandby。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组 ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-boz1qhmk
InstanceIds.N	是	Array of String	运行中状态实例列表，不支持传入非运行中状态实例。可以通过以下方式获取可用的实例ID： <ul style="list-style-type: none">通过登录控制台查询实例ID。通过调用接口 DescribeInstances，取返回信息中的 InstanceId 获取实例ID。 示例值：["ins-cri8d02t","ins-osckfnm7"]

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID。 示例值：asa-q59zikez
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 伸缩组内实例进入备用中状态

伸缩组 asg-boz1qhmk 中实例 ins-osckfnm7, ins-cri8d02t 进入备用中状态

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: EnterStandby
<公共请求参数>

{
  "AutoScalingGroupId": "asg-boz1qhmk",
  "InstanceIds": [
    "ins-osckfnm7",
    "ins-cri8d02t"
  ]
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-q59zikez",
    "RequestId": "5b039ee6-e8ff-4605-bb24-b45337747431"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalServerError	内部错误。
InternalServerError.CalleeError	调用其他服务异常。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.InstancesNotFound	指定的实例不存在。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。

启动配置相关接口

查询启动配置

最近更新时间：2025-04-25 01:08:49

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeLaunchConfigurations）用于查询启动配置的信息。

- 可以根据启动配置ID、启动配置名称等信息来查询启动配置的详细信息。过滤信息详情请见过滤器Filter。
- 如果参数为空，返回当前用户一定数量（Limit所指定的数量，默认为20）的启动配置。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeLaunchConfigurations。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LaunchConfigurationIds.N	否	Array of String	按照一个或者多个启动配置ID查询。启动配置ID形如：asc-ouy1ax38。每次请求的上限为100。参数不支持同时指定LaunchConfigurationIds和Filters 示例值：["asc-fa28v4in"]
Filters.N	否	Array of Filter	过滤条件。 <ul style="list-style-type: none">launch-configuration-id - String - 是否必填：否 - （过滤条件）按照启动配置ID过滤。launch-configuration-name - String - 是否必填：否 - （过滤条件）按照启动配置名称过滤。vague-launch-configuration-name - String - 是否必填：否 - （过滤条件）按照启动配置名称模糊搜索。tag-key - String - 是否必填：否 - （过滤条件）按照标签键进行过滤。tag-value - String - 是否必填：否 - （过滤条件）按照标签值进行过滤。tag:tag-key - String - 是否必填：否 - （过滤条件）按照标签键值对进行过滤。tag-key使用具体的标签键进行替换。使用请参考示例3 每次请求的Filters的上限为10，Filter.Values的上限为5。参数不支持同时指定LaunchConfigurationIds和Filters。 示例值：[{"Name": "launch-configuration-name", "Values": ["DataEngine123"]}]
Limit	否	Integer	返回数量，默认为20，最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值：20
Offset	否	Integer	偏移量，默认为0。关于Offset的更进一步介绍请参考 API 简介 中的相关小节。

参数名称	必选	类型	描述
			示例值: 0

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的启动配置数量。 示例值: 1
LaunchConfigurationSet	Array of LaunchConfiguration	启动配置详细信息列表。
RequestId	String	唯一请求 ID, 由服务端生成, 每次请求都会返回 (若请求因其他原因未能抵达服务端, 则该次请求不会获得 RequestId)。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 使用Filters查看启动配置列表

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeLaunchConfigurations
<公共请求参数>

{
  "Filters": [
    {
      "Values": [
        "asc-fa28v4in"
      ],
      "Name": "launch-configuration-id"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "LaunchConfigurationSet": [
      {
        "ProjectId": 0,
        "LaunchConfigurationId": "asc-fa28v4in",
        "VersionNumber": 2,
        "LaunchConfigurationName": "lc1",
        "LaunchConfigurationStatus": "NORMAL",
        "AutoScalingGroupAbstractSet": [],
        "InstanceType": "S3.MEDIUM4",
        "InstanceTypes": [
```

```
"S3.MEDIUM4"
],
"LastOperationInstanceTypesCheckPolicy": "ANY",
"ImageId": "img-eb30mz89",
"ImageFamily": "ubuntu-f0",
"Tags": [],
"IPv6InternetAccessible": {
  "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
  "InternetMaxBandwidthOut": 0,
  "BandwidthPackageId": null
},
"DisasterRecoverGroupIds": [],
"InstanceChargeType": "POSTPAID_BY_HOUR",
"InstanceMarketOptions": null,
"DiskTypePolicy": "ORIGINAL",
"SystemDisk": {
  "DiskType": "CLOUD_PREMIUM",
  "DiskSize": 50
},
>DataDisks": [
  {
    "DiskType": "CLOUD_PREMIUM",
    "DiskSize": 10,
    "SnapshotId": null,
    "DeleteWithInstance": false,
    "Encrypt": false,
    "BurstPerformance": null,
    "ThroughputPerformance": null
  }
],
>LoginSettings": {
  "KeyIds": []
},
"InternetAccessible": {
  "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
  "InternetMaxBandwidthOut": 0,
  "PublicIpAssigned": false,
  "BandwidthPackageId": null
},
"SecurityGroupIds": [],
"EnhancedService": {
  "SecurityService": {
    "Enabled": true
  },
  "AutomationToolsService": {
    "Enabled": true
  },
  "MonitorService": {
    "Enabled": true
  }
},
"UserData": null,
"InstanceTags": [],
```

```
"CreatedTime": "2022-03-03T02:36:12Z",
"UpdatedTime": "2022-03-03T06:49:31Z",
"CamRoleName": "CVM_QcsRole",
"HostNameSettings": {
  "HostName": null,
  "HostNameSuffix": null,
  "HostNameStyle": null
},
"InstanceNameSettings": {
  "InstanceName": "ins-ubuntu-0",
  "InstanceNameSuffix": null,
  "InstanceNameStyle": "UNIQUE"
},
"InstanceChargePrepaid": {
  "Period": 0,
  "RenewFlag": "NOTIFY_AND_AUTO_RENEW"
},
"HpcClusterId": "hpc-u7y5r4e3"
}
],
"RequestId": "923dd24c-e492-4bdb-90be-1d3bd4bfe8a5"
}
}
```

示例2 根据启动配置ID查询启动配置列表

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeLaunchConfigurations
<公共请求参数>

{
  "LaunchConfigurationIds": [
    "asc-g9uwgyvx",
    "asc-fa28v4in"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "LaunchConfigurationSet": [
      {
        "ProjectId": 0,
        "LaunchConfigurationId": "asc-fa28v4in",
        "VersionNumber": 2,
        "LaunchConfigurationName": "lc1",
```



```
"LaunchConfigurationStatus": "NORMAL",
"AutoScalingGroupAbstractSet": [],
"InstanceType": "S3.MEDIUM4",
"InstanceTypes": [
  "S3.MEDIUM4"
],
"LastOperationInstanceTypesCheckPolicy": "ANY",
"ImageId": "img-eb30mz89",
"InstanceChargeType": "POSTPAID_BY_HOUR",
"InstanceMarketOptions": null,
"DiskTypePolicy": "ORIGINAL",
"SystemDisk": {
  "DiskType": "CLOUD_PREMIUM",
  "DiskSize": 50
},
"DataDisks": [
  {
    "DiskType": "CLOUD_PREMIUM",
    "DiskSize": 10,
    "BurstPerformance": null,
    "SnapshotId": null,
    "DeleteWithInstance": false,
    "Encrypt": false,
    "ThroughputPerformance": null
  }
],
>LoginSettings": {
  "KeyIds": []
},
"InternetAccessible": {
  "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
  "InternetMaxBandwidthOut": 0,
  "PublicIpAssigned": false,
  "BandwidthPackageId": null
},
"SecurityGroupIds": [],
"EnhancedService": {
  "SecurityService": {
    "Enabled": true
  },
  "AutomationToolsService": {
    "Enabled": true
  },
  "MonitorService": {
    "Enabled": true
  }
},
"UserData": null,
"InstanceTags": [],
"ImageFamily": "ubuntu-f0",
"Tags": [],
"IPv6InternetAccessible": {
  "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
```

```
"InternetMaxBandwidthOut": 0,
"BandwidthPackageId": null
},
"DisasterRecoverGroupIds": [],
"CreatedTime": "2022-03-03T02:36:12Z",
"UpdatedTime": "2022-03-03T06:49:31Z",
"CamRoleName": "CVM_QcsRole",
"HostNameSettings": {
  "HostName": null,
  "HostNameSuffix": null,
  "HostNameStyle": null
},
"InstanceNameSettings": {
  "InstanceName": "ins-ubuntu-0",
  "InstanceNameSuffix": null,
  "InstanceNameStyle": "UNIQUE"
},
"InstanceChargePrepaid": {
  "Period": 0,
  "RenewFlag": "NOTIFY_AND_AUTO_RENEW"
},
"HpcClusterId": "hpc-u7y5r4e3"
},
{
  "ProjectId": 0,
  "LaunchConfigurationId": "asc-g9uwgyvx",
  "VersionNumber": 1,
  "LaunchConfigurationName": "lc2",
  "LaunchConfigurationStatus": "NORMAL",
  "AutoScalingGroupAbstractSet": [],
  "InstanceType": "S3.MEDIUM4",
  "InstanceTypes": [
    "S3.MEDIUM4"
  ],
  "LastOperationInstanceTypesCheckPolicy": "ANY",
  "ImageId": "img-eb30mz89",
  "InstanceChargeType": "POSTPAID_BY_HOUR",
  "InstanceMarketOptions": null,
  "DiskTypePolicy": "ORIGINAL",
  "SystemDisk": {
    "DiskType": "CLOUD_PREMIUM",
    "DiskSize": 50
  },
  "DataDisks": [
    {
      "DiskType": "CLOUD_PREMIUM",
      "DiskSize": 40,
      "SnapshotId": null,
      "DeleteWithInstance": null,
      "Encrypt": null,
      "BurstPerformance": null,
      "ThroughputPerformance": null
    }
  ]
}
```

```
],
"LoginSettings": {
"KeyIds": []
},
"InternetAccessible": {
"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
"InternetMaxBandwidthOut": 0,
"PublicIpAssigned": false,
"BandwidthPackageId": null
},
"SecurityGroupIds": [],
"EnhancedService": {
"SecurityService": {
"Enabled": true
},
"AutomationToolsService": {
"Enabled": true
},
"MonitorService": {
"Enabled": true
}
},
"UserData": null,
"InstanceTags": [],
"ImageFamily": "ubuntu-f0",
"Tags": [],
"IPv6InternetAccessible": {
"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
"InternetMaxBandwidthOut": 0,
"BandwidthPackageId": null
},
"DisasterRecoverGroupIds": [],
"CreatedTime": "2022-03-02T02:25:12Z",
"UpdatedTime": "2022-03-02T02:25:12Z",
"CamRoleName": "CVM_QcsRole",
"HostNameSettings": {
"HostName": null,
"HostNameSuffix": null,
"HostNameStyle": null
},
"InstanceNameSettings": {
"InstanceName": "ins-ubuntu-0",
"InstanceNameSuffix": null,
"InstanceNameStyle": "UNIQUE"
},
"InstanceChargePrepaid": {
"Period": 0,
"RenewFlag": "NOTIFY_AND_AUTO_RENEW"
},
"HpcClusterId": "hpc-u7y5r4e3"
},
"RequestId": "0d4514d4-e277-4f0f-bc85-8b7377a71980"
```

```
}  
}
```

示例3 使用Filters通过Tag:Key方式查看启动配置列表。

输入示例

```
POST / HTTP/1.1  
Host: as.tencentcloudapi.com  
Content-Type: application/json  
X-TC-Action: DescribeLaunchConfigurations  
<公共请求参数>  
  
{  
  "Filters": [  
    {  
      "Values": [  
        "v2"  
      ],  
      "Name": "tag:k1"  
    }  
  ]  
}
```

输出示例

```
{  
  "Response": {  
    "TotalCount": 1,  
    "LaunchConfigurationSet": [  
      {  
        "ProjectId": 0,  
        "LaunchConfigurationId": "asc-0zri3ck1",  
        "VersionNumber": 1,  
        "LaunchConfigurationName": "test_tag_k1_v2",  
        "LaunchConfigurationStatus": "NORMAL",  
        "AutoScalingGroupAbstractSet": [],  
        "InstanceType": "S5.MEDIUM4",  
        "InstanceTypes": [  
          "S5.MEDIUM4"  
        ],  
        "LastOperationInstanceTypesCheckPolicy": "ANY",  
        "ImageId": "img-eb30mz89",  
        "InstanceChargeType": "POSTPAID_BY_HOUR",  
        "InstanceMarketOptions": null,  
        "DiskTypePolicy": "ORIGINAL",  
        "SystemDisk": {  
          "DiskType": "CLOUD_PREMIUM",  
          "DiskSize": 50  
        },  
        "DataDisks": [  
          {  
            "DiskType": "CLOUD_PREMIUM",  
            "DiskSize": 50  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
"DiskType": "CLOUD_PREMIUM",
"DiskSize": 40,
"BurstPerformance": null,
"SnapshotId": null,
>DeleteWithInstance": null,
"Encrypt": null,
"ThroughputPerformance": null
}
],
>LoginSettings": {
  "KeyIds": []
},
"InternetAccessible": {
  "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
  "InternetMaxBandwidthOut": 0,
  "PublicIpAssigned": false,
  "BandwidthPackageId": null
},
"SecurityGroupIds": [],
"EnhancedService": {
  "SecurityService": {
    "Enabled": true
  },
  "AutomationToolsService": {
    "Enabled": true
  },
  "MonitorService": {
    "Enabled": true
  }
},
"UserData": null,
"Tags": [],
"InstanceTags": [],
"ImageFamily": "ubuntu-f0",
"IPv6InternetAccessible": {
  "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
  "InternetMaxBandwidthOut": 0,
  "BandwidthPackageId": null
},
"DisasterRecoverGroupIds": [],
"CreatedTime": "2022-07-14T06:37:08Z",
"UpdatedTime": "2022-07-14T06:37:08Z",
"CamRoleName": "CVM_QcsRole",
"HostNameSettings": {
  "HostName": null,
  "HostNameSuffix": null,
  "HostNameStyle": null
},
"InstanceNameSettings": {
  "InstanceName": "ins-ubuntu-0",
  "InstanceNameSuffix": null,
  "InstanceNameStyle": "UNIQUE"
},
}
```

```
"InstanceChargePrepaid": {
  "Period": 0,
  "RenewFlag": "NOTIFY_AND_AUTO_RENEW"
},
"HpcClusterId": "hpc-u7y5r4e3"
},
],
"RequestId": "8b8047bb-1372-4208-866e-a18e7b7547e9"
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidLaunchConfiguration	无效的启动配置。
InvalidLaunchConfigurationId	启动配置ID无效。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterConflict	指定的两个参数冲突，不能同时存在。
InvalidParameterValue.InvalidFilter	过滤条件无效。
InvalidParameterValue.InvalidLaunchConfigurationId	启动配置ID无效。
InvalidParameterValue.TooLong	取值过多。
InvalidPermission	账户不支持该操作。
LimitExceeded.FilterValuesTooLong	特定过滤器的值过多。

错误码	描述
UnsupportedOperation	操作不支持。

删除启动配置

最近更新時間：2025-05-16 01:09:32

1. 接口描述

接口請求域名：as.tencentcloudapi.com。

本接口（DeleteLaunchConfiguration）用於刪除啟動配置。

- 若啟動配置在伸縮組中屬於生效狀態，則該啟動配置不允許刪除。

默認接口請求頻率限制：10次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數 ，本接口取值：DeleteLaunchConfiguration。
Version	是	String	公共參數 ，本接口取值：2018-04-19。
Region	是	String	公共參數 ，詳見產品支持的 地域列表 。
LaunchConfigurationId	是	String	需要刪除的啟動配置ID。可通過登錄 控制台 或調用接口 DescribeLaunchConfigurations ，取返回信息中的 LaunchConfigurationId 獲取啟動配置ID。 示例值：asc-deuy6tre

3. 输出参数

參數名稱	類型	描述
RequestId	String	唯一請求 ID，由服務端生成，每次請求都會返回（若請求因其他原因未能抵達服務端，則該次請求不會獲得 RequestId）。定位問題時需要提供該次請求的 RequestId。

4. 示例

示例1 删除启动配置

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteLaunchConfiguration
<公共请求参数>

{
  "LaunchConfigurationId": "asc-fdz8j7dh"
}
```

输出示例


```
{
  "Response": {
    "RequestId": "dfc4da00-b643-405c-bb4f-d590cfa48b92"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidLaunchConfigurationId	启动配置ID无效。
ResourceInUse.LaunchConfigurationIdInUse	指定的启动配置仍在伸缩组中使用。
ResourceNotFound.LaunchConfigurationIdNotFound	指定的启动配置不存在。

创建启动配置

最近更新时间：2025-05-16 01:09:33

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（CreateLaunchConfiguration）用于创建新的启动配置。

- 启动配置，可以通过 [ModifyLaunchConfigurationAttributes](#) 修改少量字段。如需使用新的启动配置，建议重新创建启动配置。
- 每个地域默认只能创建50个启动配置，详见[使用限制](#)。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateLaunchConfiguration。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LaunchConfigurationName	是	String	启动配置显示名称。名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点，最大长度不能超60个字节。 示例值：as_test
ImageId	否	String	指定有效的 镜像ID ，格式形如img-8toqc6s3。镜像Id与镜像族名称，二者必填一个且只能填写一个。镜像类型分为四种： <ul style="list-style-type: none">公共镜像自定义镜像共享镜像服务市场镜像 可通过以下方式获取可用的镜像ID： <ul style="list-style-type: none">公共镜像、自定义镜像、共享镜像的镜像ID可通过登录控制台查询；服务镜像市场的镜像ID可通过云市场查询。通过调用接口 DescribeImages，取返回信息中的ImageId字段。 示例值：img-8toqc6s3
ProjectId	否	Integer	启动配置所属项目ID。默认为0，表示使用默认项目。该参数可以通过调用 DescribeProject 的返回值中的 projectId 字段来获取。 注意：伸缩组内实例所属项目ID取伸缩组项目ID，与这里取值无关。 示例值：0
InstanceType	否	String	实例机型。不同实例机型指定了不同的资源规格，具体取值可通过调用接口 DescribeInstanceTypeConfigs 来获得最新的规格

参数名称	必选	类型	描述
			表或参见 实例类型描述 。 InstanceType和InstanceTypes参数互斥，二者必填一个且只能填写一个。 示例值：S2.SMALL1
SystemDisk	否	SystemDisk	实例系统盘配置信息。若不指定该参数，则按照系统默认值进行分配。 示例值：{"DiskSize": 500, "DiskType": "CLOUD_BSSD"}
DataDisks.N	否	Array of DataDisk	实例数据盘配置信息。若不指定该参数，则默认不购买数据盘，最多支持指定11块数据盘。 示例值：[{"DiskSize": 200, "DiskType": "CLOUD_BSSD"}]
InternetAccessible	否	InternetAccessible	公网带宽相关信息设置。若不指定该参数，则默认公网带宽为0Mbps。 示例值：{"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR", "InternetMaxBandwidthOut": 0, "PublicIpAssigned": false}
LoginSettings	否	LoginSettings	实例登录设置。通过该参数可以设置实例的登录方式密码、密钥或保持镜像的原始登录设置。默认情况下会随机生成密码，并以站内信方式知会到用户。 示例值：{"KeyIds": ["skey-alaz9k65"]}
SecurityGroupIds.N	否	Array of String	实例所属安全组。该参数可以通过调用 DescribeSecurityGroups 的返回值中的SecurityGroupId字段来获取。若不指定该参数，则默认不绑定安全组。 示例值：["sg-arrfspxo"]
EnhancedService	否	EnhancedService	增强服务。通过该参数可以指定是否开启云安全、云监控等服务。若不指定该参数，则默认开启云监控、云安全服务。 示例值：{"MonitorService": {"Enabled": true}, "SecurityService": {"Enabled": true}}
UserData	否	String	经过 Base64 编码后的自定义数据，最大长度不超过16KB。 示例值：lyEvYmluL2Jhc2gKZWNoYXxMjMKCg==
InstanceChargeType	否	String	实例计费类型，CVM默认值按照POSTPAID_BY_HOUR处理。 <ul style="list-style-type: none"> POSTPAID_BY_HOUR：按小时后付费 SPOTPAID：竞价付费 PREPAID：预付费，即包年包月 CDCPAID：专用集群付费 示例值：POSTPAID_BY_HOUR
InstanceMarketOptions	否	InstanceMarketOptionsRequest	实例的市场相关选项，如竞价实例相关参数，若指定实例的付费模式为竞价付费则该参数必传。 示例值：{"SpotOptions": {"MaxPrice": 1000}}
InstanceTypes.N	否	Array of String	实例机型列表，不同实例机型指定了不同的资源规格，最多支持10种实例机型。 InstanceType和InstanceTypes参数互斥，二者必填一个且只能填写一个。具体取值可通过调用接口 DescribeInstanceTypeConfigs 来获得最新的规格表或参见 实例规格描述 。 示例值：["S5.MEDIUM2", "S6.MEDIUM2"]
CamRoleName	否	String	CAM角色名称。可通过 DescribeRoleList 接口返回值中的roleName获取。

参数名称	必选	类型	描述
			示例值: CVM_QcsRole
InstanceTypesCheckPolicy	否	String	实例类型校验策略, 取值包括 ALL 和 ANY, 默认取值为ANY。该参数仅在 InstanceTypes 入参包含多个机型时生效。 <ul style="list-style-type: none"> ALL, 所有实例类型 (InstanceType) 都可用则通过校验, 否则校验报错。 ANY, 存在任何一个实例类型 (InstanceType) 可用则通过校验, 否则校验报错。 实例类型不可用的常见原因包括该实例类型售罄、对应云盘售罄等。 如果 InstanceTypes 中一款机型不存在或者已下线, 则无论 InstanceTypesCheckPolicy 采用何种取值, 都会校验报错。 示例值: ANY
InstanceTags.N	否	Array of InstanceTag	标签列表。通过指定该参数, 可以为扩容的实例绑定标签。最多支持指定10个标签。 示例值: [{"Key": "server-group-name", "Value": "as-tencent"}]
Tags.N	否	Array of Tag	标签描述列表。通过指定该参数可以支持绑定标签到启动配置。每个启动配置最多支持30个标签。 示例值: [{"Key": "CCode", "Value": "ASTENCENT", "ResourceType": "launch-configuration"}]
HostNameSettings	否	HostNameSettings	云服务器主机名 (HostName) 的相关设置。 示例值: {"HostName": "app-as", "HostNameStyle": "UNIQUE"}
InstanceNameSettings	否	InstanceNameSettings	云服务器实例名 (InstanceName) 的相关设置。 如果用户在启动配置中设置此字段, 则伸缩组创建出的实例 InstanceName 参照此字段进行设置, 并传递给 CVM; 如果用户未在启动配置中设置此字段, 则伸缩组创建出的实例 InstanceName 按照 “as-{{ 伸缩组 AutoScalingGroupName }}” 进行设置, 并传递给 CVM。 示例值: {"InstanceName": "app-as", "InstanceNameStyle": "UNIQUE"}
InstanceChargePrepaid	否	InstanceChargePrepaid	预付费模式, 即包年包月相关参数设置。通过该参数可以指定包年包月实例的购买时长、是否设置自动续费等属性。若指定实例的付费模式为预付费则该参数必传。 示例值: {"Period": 1, "RenewFlag": "NOTIFY_AND_AUTO_RENEW"}
DiskTypePolicy	否	String	云盘类型选择策略, 默认取值 ORIGINAL, 取值范围: <ul style="list-style-type: none"> ORIGINAL: 使用设置的云盘类型 AUTOMATIC: 自动选择当前可用的云盘类型 示例值: ORIGINAL
HpcClusterId	否	String	高性能计算集群ID。可通过调用 DescribeHpcClusters 接口获取该参数。 注意: 此字段默认为空。 示例值: hpc-gt4es2q1
IPv6InternetAccessible	否	IPv6InternetAccessible	IPv6公网带宽相关信息设置。若新建实例包含IPv6地址, 该参数可为新建实例的IPv6地址分配公网带宽。关联启动配置的伸缩组 Ipv6AddressCount参数为0时, 该参数不会生效。 示例值: {"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR", "InternetMaxBandwidthOut": 1}

参数名称	必选	类型	描述
DisasterRecoverGroupIds.N	否	Array of String	置放群组id, 仅支持指定一个。 示例值: ["ps-dy7ui89o"]
ImageFamily	否	String	镜像族名称。镜像Id与镜像族名称, 二者必填一个且只能填写一个。可通过调用 DescribeImages 接口获取该参数。 示例值: test-family
DedicatedClusterId	否	String	本地专用集群ID。可通过 DescribeDedicatedClusters 接口获取该参数。 示例值: cluster-gbo27yc4
Metadata	否	Metadata	自定义metadata。

3. 输出参数

参数名称	类型	描述
LaunchConfigurationId	String	当通过本接口来创建启动配置时会返回该参数, 表示启动配置ID。 示例值: asc-23h37kyn
RequestId	String	唯一请求 ID, 由服务端生成, 每次请求都会返回 (若请求因其他原因未能抵达服务端, 则该次请求不会获得 RequestId)。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 简单参数创建

只传必传的启动配置名称, 实例机型, 镜像ID, 其他均采用系统默认值, 具体配置如下: 启动配置名称为: as_test, 实例机型为: 标准2型 1C1G (S2.SMALL1), 镜像ID为: img-8toqc6s3。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateLaunchConfiguration
<公共请求参数>

{
  "ImageId": "img-8toqc6s3",
  "InstanceType": "S2.SMALL1",
  "LaunchConfigurationName": "as_test"
}
```

输出示例

```
{
  "Response": {
    "LaunchConfigurationId": "asc-23h37kyn",
    "RequestId": "d639dd64-9e46-4246-b13c-80954f81c11b"
  }
}
```

示例2 详细参数创建

启动配置名称为：as_test，镜像ID为：img-8toqc6s3，选择机型为：标准2型 1C1G（S2.SMALL1），系统盘选择 50G 本地硬盘，数据盘选择100G 普通云硬盘，实例销毁时不保留数据盘，数据盘不加密，公网付费模式为流量按小时后付费，外网带宽上限 5 Mbps，分配公网IP，选择密钥登录，安装云监控云安全。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateLaunchConfiguration
<公共请求参数>

{
  "SystemDisk": {
    "DiskSize": "50",
    "DiskType": "LOCAL_BASIC"
  },
  "LoginSettings": {
    "KeyIds": [
      "skey-k8eypc11"
    ]
  },
  "ImageId": "img-8toqc6s3",
  "EnhancedService": {
    "SecurityService": {
      "Enabled": "TRUE"
    },
    "MonitorService": {
      "Enabled": "TRUE"
    }
  },
  "LaunchConfigurationName": "as_test",
  "InternetAccessible": {
    "PublicIpAssigned": "TRUE",
    "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
    "InternetMaxBandwidthOut": "5"
  },
  "InstanceType": "S2.SMALL1",
  "DataDisks": [
    {
      "Encrypt": "FALSE",
      "DeleteWithInstance": "TRUE",
      "DiskSize": "100",
      "DiskType": "CLOUD_BASIC"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "LaunchConfigurationId": "asc-fdz8j7dh",
```

```
"RequestId": "9a7209d3-2260-49d7-952a-dfa2001f8822"
}
```

示例3 创建竞价实例配置

启动配置名称为：spot-test，机型为：标准2型 2C4G（S2.MEDIUM4），计费配置为竞价（SPOTPAID），最高出价为0.99元/小时。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateLaunchConfiguration
<公共请求参数>

{
  "SystemDisk": {
    "DiskSize": "50",
    "DiskType": "CLOUD_PREMIUM"
  },
  "InstanceMarketOptions": {
    "SpotOptions": {
      "SpotInstanceType": "one-time",
      "MaxPrice": "0.99"
    },
    "MarketType": "spot"
  },
  "ImageId": "img-8toqc6s3",
  "InstanceChargeType": "SPOTPAID",
  "LaunchConfigurationName": "spot-test",
  "InternetAccessible": {
    "PublicIpAssigned": "true",
    "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR",
    "InternetMaxBandwidthOut": "20"
  },
  "InstanceType": "S2.MEDIUM4"
}
```

输出示例

```
{
  "Response": {
    "LaunchConfigurationId": "asc-hpzwe3o2",
    "RequestId": "ccfe3052-e9c9-47ee-bf3d-5bc2dfd972c0"
  }
}
```

示例4 创建启动配置，支持多种实例机型

支持两种实例机型，分别是S2.SMALL2和S2.SMALL4

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateLaunchConfiguration
<公共请求参数>

{
  "ImageId": "img-8toqc6s3",
  "InstanceTypes": [
    "S2.SMALL4",
    "S2.SMALL2"
  ],
  "LaunchConfigurationName": "multi_instance_types"
}
```

输出示例

```
{
  "Response": {
    "LaunchConfigurationId": "asc-77mh1cho",
    "RequestId": "2864c860-27a0-439e-a1e1-0003b76734e7"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.CallStsError	STS 接口调用失败。
InternalServerError.CalleeError	调用其他服务异常。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.HostNameUnavailable	主机名参数不适用于该镜像。
InvalidParameter.InvalidCombination	无效的参数组合。
InvalidParameter.MustOneParameter	参数缺失，两种参数之中必须指定其中一个。
InvalidParameter.ParameterDeprecated	该参数已废弃。
InvalidParameter.ParameterMustBeDeleted	部分参数存在互斥应该删掉。
InvalidParameterValue.AccountNotSupportBandwidthPackageId	当前账户不支持带宽包ID参数。
InvalidParameterValue.CvmConfigurationError	CVM参数校验异常。
InvalidParameterValue.HostNameIllegal	主机名不合法。
InvalidParameterValue.HostNameWithSuffixTooLong	带后缀的主机名称过长，超过了规定上限。
InvalidParameterValue.IPv6InternetChargeType	指定的IPv6公网带宽计费模式不合法。
InvalidParameterValue.ImageNotFound	指定的镜像不存在。
InvalidParameterValue.InstanceNameIllegal	设置的实例名称不合法。
InvalidParameterValue.InstanceNameWithSuffixTooLong	带后缀的实例名称过长，超过了规定上限。
InvalidParameterValue.InstanceTypeNotSupported	实例机型不支持。
InvalidParameterValue.InvalidDisasterRecoverGroupId	指定的置放群组ID格式不正确。
InvalidParameterValue.InvalidHpcClusterId	高性能计算集群ID无效。
InvalidParameterValue.InvalidImageId	镜像ID无效。
InvalidParameterValue.InvalidInstanceType	实例机型无效。
InvalidParameterValue.InvalidLaunchConfiguration	输入的启动配置无效。
InvalidParameterValue.InvalidSecurityGroupId	安全组ID无效。
InvalidParameterValue.InvalidSnapshotId	快照ID无效。
InvalidParameterValue.LaunchConfigurationNameDuplicated	启动配置名称重复。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.MissingBandwidthPackageId	当前账户在带宽包模式中必须填写带宽包ID参数。
InvalidParameterValue.NotStringTypeFloat	提供的值不是浮点字符串格式。
InvalidParameterValue.ProjectIdNotFound	项目ID不存在。
InvalidParameterValue.TooLong	取值过多。
InvalidParameterValue.TooShort	输入参数值的长度小于最小值。
InvalidParameterValue.UserDataFormatError	UserData格式错误。
InvalidParameterValue.UserDataSizeExceeded	UserData长度过长。

错误码	描述
InvalidPermission	账户不支持该操作。
LimitExceeded.LaunchConfigurationQuotaNotEnough	启动配置配额不足。
MissingParameter	缺少参数错误。
MissingParameter.InstanceMarketOptions	竞价计费类型缺少对应的 InstanceMarketOptions 参数。
ResourceNotFound.BandwidthPackageIdNotFound	指定的带宽包ID不存在。
ResourceNotFound.DisasterRecoverGroupNotFound	指定的置放群组ID不存在。
UnauthorizedOperation.AutoScalingRoleUnauthorized	未授权弹性伸缩预设服务角色AS_QCSRole，请前往弹性伸缩控制台进行授权。

修改启动配置属性

最近更新时间：2025-05-16 01:09:32

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ModifyLaunchConfigurationAttributes）用于修改启动配置部分属性。

- 修改启动配置后，已经使用该启动配置扩容的存量实例不会发生变更，此后使用该启动配置的新增实例会按照新的配置进行扩容。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyLaunchConfigurationAttributes。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LaunchConfigurationId	是	String	启动配置ID。可通过登录 控制台 或调用接口 DescribeLaunchConfigurations ，取返回信息中的 LaunchConfigurationId 获取启动配置ID。 示例值：asc-aaaabbbb
ImageId	否	String	指定有效的 镜像ID ，格式形如img-8toqc6s3。镜像类型分为四种： <ul style="list-style-type: none">公共镜像自定义镜像共享镜像服务市场镜像 可通过以下方式获取可用的镜像ID： <ul style="list-style-type: none">公共镜像、自定义镜像、共享镜像的镜像ID可通过登录控制台查询；服务镜像市场的镜像ID可通过云市场查询。通过调用接口 DescribeImages，取返回信息中的ImageId 字段。 示例值：img-8toqc6s3
InstanceTypes.N	否	Array of String	实例类型列表，不同实例机型指定了不同的资源规格，最多支持10种实例机型。 InstanceType 指定单一实例类型，通过设置 InstanceTypes 可以指定多实例类型，并使原有的InstanceType失效。具体取值可通过调用接口 DescribeInstanceTypeConfigs 来获得最新的规格表或参见 实例规格描述 。 示例值：S5.SMALL2
InstanceTypesCheckPolicy	否	String	实例类型校验策略，在实际修改 InstanceTypes 时发挥作用，取值包括 ALL 和 ANY，默认取值为ANY。

参数名称	必选	类型	描述
			<ul style="list-style-type: none"> ALL, 所有实例类型 (InstanceType) 都可用则通过校验, 否则校验报错。 ANY, 存在任何一个实例类型 (InstanceType) 可用则通过校验, 否则校验报错。 <p>实例类型不可用的常见原因包括该实例类型售罄、对应云盘售罄等。</p> <p>如果 InstanceTypes 中一款机型不存在或者已下线, 则无论 InstanceTypesCheckPolicy 采用何种取值, 都会校验报错。</p> <p>示例值: ALL</p>
LaunchConfigurationName	否	String	<p>启动配置显示名称。名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点, 最大长度不能超60个字节。</p> <p>示例值: myascname</p>
UserData	否	String	<p>经过 Base64 编码后的自定义数据, 最大长度不超过16KB。如果要清空UserData, 则指定其为空字符串。</p> <p>示例值: lyEvYmluL2Jhc2gKZWNoByAxMjMKCg==</p>
SecurityGroupIds.N	否	Array of String	<p>实例所属安全组。该参数可以通过调用 DescribeSecurityGroups 的返回值中的SecurityGroupId 字段来获取。</p> <p>若指定该参数, 请至少提供一个安全组, 列表顺序有先后。</p> <p>示例值: ["sg-aaaabbbb"]</p>
InternetAccessible	否	InternetAccessible	<p>公网带宽相关信息设置。</p> <p>当公网出带宽上限为0Mbps时, 不支持修改为开通分配公网IP; 相应的, 当前为开通分配公网IP时, 修改的公网出带宽上限值必须大于0Mbps。</p> <p>示例值: {"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR", "InternetMaxBandwidthOut": 0, "PublicIpAssigned": false}</p>
InstanceChargeType	否	String	<p>实例计费类型。具体取值范围如下:</p> <ul style="list-style-type: none"> POSTPAID_BY_HOUR: 按小时后付费 SPOTPAID: 竞价付费 PREPAID: 预付费, 即包年包月 CDCPAID: 专用集群付费 <p>示例值: POSTPAID_BY_HOUR</p>
InstanceChargePrepaid	否	InstanceChargePrepaid	<p>预付费模式, 即包年包月相关参数设置。通过该参数可以指定包年包月实例的购买时长、是否设置自动续费等属性。</p> <p>若修改实例的付费模式为预付费, 则该参数必传; 从预付费修改为其他付费模式时, 本字段原信息会自动丢弃。</p> <p>当新增该字段时, 必须传递购买实例的时长, 其它未传递字段会设置为默认值。</p> <p>当修改本字段时, 当前付费模式必须为预付费。</p> <p>示例值: {"Period": 1, "RenewFlag": "NOTIFY_AND_AUTO_RENEW"}</p>
InstanceMarketOptions	否	InstanceMarketOptionsRequest	<p>实例的市场相关选项, 如竞价实例相关参数。</p> <p>若修改实例的付费模式为竞价付费, 则该参数必传; 从竞价付费修改为其他付费模式时, 本字段原信息会自动丢弃。</p> <p>当新增该字段时, 必须传递竞价相关选项下的竞价出价, 其它未传递字段会设置为默认值。</p> <p>当修改本字段时, 当前付费模式必须为竞价付费。</p> <p>示例值: {"SpotOptions": {"MaxPrice": 1000}}</p>
DiskTypePolicy	否	String	<p>云盘类型选择策略, 取值范围:</p>

参数名称	必选	类型	描述
			<ul style="list-style-type: none"> ORIGINAL: 使用设置的云盘类型。 AUTOMATIC: 自动选择当前可用的云盘类型。 示例值: ORIGINAL
SystemDisk	否	SystemDisk	实例系统盘配置信息。 示例值: {"DiskSize": 500, "DiskType": "CLOUD_BSSD"}
DataDisks.N	否	Array of DataDisk	实例数据盘配置信息。 最多支持指定11块数据盘。采取整体修改, 因此请提供修改后的全部值。 数据盘类型默认与系统盘类型保持一致。 示例值: [{"DiskSize": 200, "DiskType": "CLOUD_BSSD"}]
HostNameSettings	否	HostNameSettings	云服务器主机名 (HostName) 的相关设置。 不支持windows实例设置主机名。 新增该属性时, 必须传递云服务器的主机名, 其它未传递字段会设置为默认值。 示例值: {"HostName": "app-as", "HostNameStyle": "UNIQUE"}
InstanceNameSettings	否	InstanceNameSettings	云服务器 (InstanceName) 实例名的相关设置。 如果用户在启动配置中设置此字段, 则伸缩组创建出的实例 InstanceName 参照此字段进行设置, 并传递给 CVM; 如果用户未在启动配置中设置此字段, 则伸缩组创建出的实例 InstanceName 按照 "as-{{ 伸缩组 AutoScalingGroupName }}" 进行设置, 并传递给 CVM。 新增该属性时, 必须传递云服务器的实例名称, 其它未传递字段会设置为默认值。 示例值: {"InstanceName": "app-as", "InstanceNameStyle": "UNIQUE"}
EnhancedService	否	EnhancedService	增强服务。通过该参数可以指定是否开启云安全、云监控等服务。 示例值: {"MonitorService": {"Enabled": true}, "SecurityService": {"Enabled": true}}
CamRoleName	否	String	CAM角色名称。可通过 DescribeRoleList 接口返回值中的 roleName 获取。 示例值: TestRole
HpcClusterId	否	String	高性能计算集群ID。可通过调用 DescribeHpcClusters 接口获取该参数。 注意: 此字段默认为空。 示例值: hpc-abcd1234
IPv6InternetAccessible	否	IPv6InternetAccessible	IPv6公网带宽相关信息设置。若新建实例包含IPv6地址, 该参数可为新建实例的IPv6地址分配公网带宽。关联启动配置的伸缩组 Ipv6AddressCount 参数为0时, 该参数不会生效。 示例值: {"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR", "InternetMaxBandwidthOut": 1}
DisasterRecoverGroupIds.N	否	Array of String	置放群组id, 仅支持指定一个。可通过调用 DescribeDisasterRecoverGroups 接口获取该参数。 示例值: ["ps-fy7e0ml0"]
LoginSettings	否	LoginSettings	实例登录设置, 包括密码、密钥或保持镜像的原始登录设置。 请注意, 指定新的登录设置会覆盖原有登录设置。例如, 如果您之前使用密码登录, 使用该参数将登录设置修改为密钥, 则原有密码被清除。 示例值: {"KeyIds": ["skey-alaz9k65"]}

参数名称	必选	类型	描述
InstanceTags.N	否	Array of InstanceTag	实例标签列表。通过指定该参数，可以为扩容的实例绑定标签。最多支持指定10个标签。 该参数会覆盖原有的实例标签列表，如需新增标签，需将新标签和原有标签一并传入。 示例值：[{"Key":"Test","Value":"test"}]
ImageFamily	否	String	镜像族名称。可通过调用 DescribeImages 接口获取该参数。 示例值：test-family
DedicatedClusterId	否	String	本地专用集群ID。 示例值：cluster-gbo27yc4
Metadata	否	Metadata	自定义metadata。

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 清空UserData

指定启动配置asc-291kq6ku，将UserData修改为空字符串

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyLaunchConfigurationAttributes
<公共请求参数>

{
  "LaunchConfigurationId": "asc-291kq6ku",
  "UserData": ""
}
```

输出示例

```
{
  "Response": {
    "RequestId": "2c027f22-3a3b-489a-a77a-89c53fc15212"
  }
}
```

示例2 指定启动配置，修改镜像、实例类型、名称

指定启动配置asc-291kq6ku，修改镜像为img-8toqc6s3，修改实例类型为S2.SMALL1，修改启动配置名称为updated_config

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyLaunchConfigurationAttributes
```

<公共请求参数>

```
{
  "ImageId": "img-8toqc6s3",
  "InstanceTypes": [
    "S2.SMALL1"
  ],
  "LaunchConfigurationName": "updated_config",
  "LaunchConfigurationId": "asc-291kq6ku"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "07022dcb-5bba-48f0-a2b0-800ad006d031"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.CalleeError	调用其他服务异常。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.HostNameUnavailable	主机名参数不适用于该镜像。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameter.InvalidCombination	无效的参数组合。
InvalidParameter.ParameterDeprecated	该参数已废弃。
InvalidParameter.ParameterMustBeDeleted	部分参数存在互斥应该删掉。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.AccountNotSupportBandwidthPackageId	当前账户不支持带宽包ID参数。
InvalidParameterValue.CvmConfigurationError	CVM参数校验异常。
InvalidParameterValue.HostNameIllegal	主机名不合法。
InvalidParameterValue.HostNameWithSuffixTooLong	带后缀的主机名称过长，超过了规定上限。
InvalidParameterValue.IPV6InternetChargeType	指定的IPv6公网带宽计费模式不合法。
InvalidParameterValue.ImageNotFound	指定的镜像不存在。
InvalidParameterValue.InstanceNameIllegal	设置的实例名称不合法。
InvalidParameterValue.InstanceNameWithSuffixTooLong	带后缀的实例名称过长，超过了规定上限。
InvalidParameterValue.InstanceTypeNotSupported	实例机型不支持。
InvalidParameterValue.InvalidDisasterRecoverGroupId	指定的置放群组ID格式不正确。
InvalidParameterValue.InvalidHpcClusterId	高性能计算集群ID无效。
InvalidParameterValue.InvalidImageId	镜像ID无效。
InvalidParameterValue.InvalidInstanceType	实例机型无效。
InvalidParameterValue.InvalidLaunchConfigurationId	启动配置ID无效。
InvalidParameterValue.InvalidSecurityGroupId	安全组ID无效。
InvalidParameterValue.LaunchConfigurationNameDuplicated	启动配置名称重复。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.MissingBandwidthPackageId	当前账户在带宽包模式中必须填写带宽包ID参数。
InvalidParameterValue.NotStringTypeFloat	提供的值不是浮点字符串格式。
InvalidParameterValue.TooLong	取值过多。
InvalidParameterValue.TooShort	输入参数值的长度小于最小值。
InvalidParameterValue.UserDataFormatError	UserData格式错误。
InvalidParameterValue.UserDataSizeExceeded	UserData长度过长。
MissingParameter	缺少参数错误。
MissingParameter.InScenario	在特定场景下缺少参数。

错误码	描述
ResourceNotFound.BandwidthPackageIdNotFound	指定的带宽包ID不存在。
ResourceNotFound.DisasterRecoverGroupNotFound	指定的置放群组ID不存在。
ResourceNotFound.LaunchConfigurationIdNotFound	指定的启动配置不存在。

清除启动配置属性

最近更新时间：2025-04-28 01:09:37

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ClearLaunchConfigurationAttributes）用于将启动配置内的特定属性完全清空。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ClearLaunchConfigurationAttributes。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LaunchConfigurationId	是	String	启动配置ID。可通过如下方式获取： <ul style="list-style-type: none">通过登录 控制台 查询启动配置ID。通过调用接口 DescribeLaunchConfigurations，取返回信息中的 LaunchConfigurationId 获取启动配置ID。 示例值：asc-w37yt5r4
ClearDataDisks	否	Boolean	是否清空数据盘信息，非必填，默认为 false。 填 true 代表清空“数据盘”信息，清空后基于此新创建的云主机将不含有任何数据盘。 示例值：false
ClearHostNameSettings	否	Boolean	是否清空云服务器主机名相关设置信息，非必填，默认为 false。 填 true 代表清空主机名设置信息，清空后基于此新创建的云主机将不设置主机名。 示例值：false
ClearInstanceNameSettings	否	Boolean	是否清空云服务器实例名相关设置信息，非必填，默认为 false。 填 true 代表清空主机名设置信息，清空后基于此新创建的云主机将按照“as-{{ 伸缩组 AutoScalingGroupName }}”进行设置。 示例值：false
ClearDisasterRecoverGroupIds	否	Boolean	是否清空置放群组信息，非必填，默认为 false。 填 true 代表清空置放群组信息，清空后基于此新创建的云主机将不指定任何置放群组。 示例值：false
ClearInstanceTags	否	Boolean	是否清空实例标签列表，非必填，默认为 false。 填 true 代表清空实例标签列表，清空后基于此新创建的云主机将不会绑定列表中的标签。 示例值：false
ClearMetadata	否	Boolean	是否清空 MetaData，非必填，默认为 false。填 true 代表清空 MetaData，清空后基于此新创建的云主机将不会关联自定义的 Metadata。 示例值：false

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 清空启动配置中的数据盘项

指定启动配置asc-kr4beurf，入参ClearDataDisks设置为true，接口调用后启动配置中的数据盘信息被清空，后续基于asc-kr4beurf创建的云主机将不再分配任何数据盘

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ClearLaunchConfigurationAttributes
<公共请求参数>

{
  "LaunchConfigurationId": "asc-kr4beurf",
  "ClearDataDisks": true
}
```

输出示例

```
{
  "Response": {
    "RequestId": "382c6cad-15ae-496a-a965-66b95674f5a7"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidLaunchConfigurationId	启动配置ID无效。

定时任务相关接口

修改定时任务

最近更新时间：2025-04-28 01:09:36

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ModifyScheduledAction）用于修改定时任务。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyScheduledAction。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ScheduledActionId	是	String	待修改的定时任务ID。可以通过调用接口 DescribeScheduledActions ，取返回信息中的 ScheduledActionId 获取定时任务ID。 示例值：asst-chwbkq4c
ScheduledActionName	否	String	定时任务名称。名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点，最大长度不能超过60个字节。同一伸缩组下必须唯一。 示例值：scheduled-action-0
MaxSize	否	Integer	当定时任务触发时，设置的伸缩组最大实例数。 示例值：5
MinSize	否	Integer	当定时任务触发时，设置的伸缩组最小实例数。 示例值：0
DesiredCapacity	否	Integer	当定时任务触发时，设置的伸缩组期望实例数。 示例值：3
StartTime	否	Timestamp ISO8601	定时任务的首次触发时间，取值为北京时间（UTC+8），按照ISO8601标准，格式：YYYY-MM-DDThh:mm:ss+08:00。 示例值：2018-08-28T23:00:00+08:00
EndTime	否	Timestamp ISO8601	定时任务的结束时间，取值为北京时间（UTC+8），按照ISO8601标准，格式：YYYY-MM-DDThh:mm:ss+08:00。 此参数与Recurrence需要同时指定，到达结束时间之后，定时任务将不再生效。 示例值：2019-01-01T00:00:00+08:00
Recurrence	否	String	定时任务的重复方式。为标准 Cron 格式， Recurrence参数限制 为5个字段，由空格分开，结构为：分，小时，日期，月份，星期。此参数与EndTime需要同时指定。 示例值：0 18 * * 1,2,3,4,5

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改定时任务

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyScheduledAction
<公共请求参数>

{
  "DesiredCapacity": 3,
  "MinSize": 0,
  "MaxSize": 5,
  "ScheduledActionName": "scheduled-action-0",
  "StartTime": "2018-08-28T23:00:00+08:00",
  "ScheduledActionId": "asst-chwbkq4c"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "5f6f0f95-216f-4745-a2e6-617897e9cedb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.CronExpressionIllegal	定时任务指定的Cron表达式无效。
InvalidParameterValue.EndTimeBeforeStartTime	定时任务设置的结束时间在开始时间。
InvalidParameterValue.InvalidScheduledActionId	定时任务ID无效。
InvalidParameterValue.InvalidScheduledActionNameIncludeIllegalChar	定时任务名称包含无效字符。
InvalidParameterValue.ScheduledActionNameDuplicate	定时任务名称重复。
InvalidParameterValue.Size	伸缩组最大数量、最小数量、期望实例数取值不合法。
InvalidParameterValue.StartTimeBeforeCurrentTime	定时任务设置的开始时间在当前时间之前。
InvalidParameterValue.TimeFormat	时间格式错误。
InvalidParameterValue.TooLong	取值过多。
LimitExceeded.DesiredCapacityLimitExceeded	期望实例数超出限制。
LimitExceeded.MaxSizeLimitExceeded	最大实例数大于限制。
LimitExceeded.MinSizeLimitExceeded	最小实例数低于限制。
LimitExceeded.ScheduledActionLimitExceeded	定时任务数量超过限制。
ResourceNotFound.ScheduledActionNotFound	指定的定时任务不存在。

查询定时任务

最近更新时间：2025-04-28 01:09:36

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口 (DescribeScheduledActions) 用于查询一个或多个定时任务的详细信息。

- 可以根据定时任务ID、定时任务名称或者伸缩组ID等信息来查询定时任务的详细信息。过滤信息详细请见过滤器Filter。
- 如果参数为空，返回当前用户一定数量（Limit所指定的数量，默认为20）的定时任务。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeScheduledActions。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ScheduledActionIds.N	否	Array of String	按照一个或者多个定时任务ID查询。可通过登录 控制台 获取定时任务 ID。每次请求的实例的上限为100。参数不支持同时指定ScheduledActionIds和Filters。 示例值：["asst-caa5ha40"]
Filters.N	否	Array of Filter	过滤条件。可通过登录 控制台 获取定时任务 ID，定时任务名称，伸缩组 ID。 <ul style="list-style-type: none">scheduled-action-id - String - 是否必填：否 - （过滤条件）按照定时任务ID过滤。scheduled-action-name - String - 是否必填：否 - （过滤条件）按照定时任务名称过滤。auto-scaling-group-id - String - 是否必填：否 - （过滤条件）按照伸缩组ID过滤。 示例值：[{"Name": "auto-scaling-group-name", "Values": ["production-as"]}]
Offset	否	Integer	偏移量，默认为0。关于Offset的更进一步介绍请参考 API 简介 中的相关小节。 示例值：0
Limit	否	Integer	返回数量，默认为20，最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值：20

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的定时任务数量。 示例值：1
ScheduledActionSet	Array of ScheduledAction	定时任务详细信息列表。

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询定时任务

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeScheduledActions
<公共请求参数>

{
  "ScheduledActionIds": [
    "asst-caa5ha40"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "cc207181-288d-4a39-a1d1-63a5e1ba1d2b",
    "TotalCount": 1,
    "ScheduledActionSet": [
      {
        "ScheduledActionId": "asst-jf898dps",
        "ScheduledActionName": "product-a0-sche0",
        "AutoScalingGroupId": "asg-keqt9eg1",
        "StartTime": "2022-02-22T16:00:00+08:00",
        "Recurrence": "0 0 * * *",
        "EndTime": "2024-02-22T16:00:00+08:00",
        "MaxSize": 15,
        "DesiredCapacity": 1,
        "MinSize": 1,
        "CreatedTime": "2022-02-21T02:19:52Z",
        "ScheduledType": "CRONTAB"
      }
    ]
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue.Filter	无效的过滤器。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidFilter	过滤条件无效。
InvalidParameterValue.InvalidScheduledActionId	定时任务ID无效。
ResourceNotFound.ScheduledActionNotFound	指定的定时任务不存在。

删除定时任务

最近更新时间：2025-04-28 01:09:36

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DeleteScheduledAction）用于删除特定的定时任务。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteScheduledAction。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ScheduledActionId	是	String	待删除的定时任务ID。可以通过调用接口 DescribeScheduledActions ，取返回信息中的 ScheduledActionId 获取定时任务ID。 示例值：asst-chwbkq4c

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除定时任务

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteScheduledAction
<公共请求参数>

{
  "ScheduledActionId": "asst-chwbkq4c"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "cde28f43-67f6-4084-a01e-9623c08ff288"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidScheduledActionId	定时任务ID无效。
ResourceNotFound.ScheduledActionNotFound	指定的定时任务不存在。

创建定时任务

最近更新时间：2025-04-28 01:09:37

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（CreateScheduledAction）用于创建定时任务。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateScheduledAction。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-2nr9xh8h
ScheduledActionName	是	String	定时任务名称。名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点，最大长度不能超过60个字节。同一伸缩组下必须唯一。 示例值：scheduled-action-0
MaxSize	是	Integer	当定时任务触发时，设置的伸缩组最大实例数。 示例值：10
MinSize	是	Integer	当定时任务触发时，设置的伸缩组最小实例数。 示例值：4
DesiredCapacity	是	Integer	当定时任务触发时，设置的伸缩组期望实例数。 示例值：6
StartTime	是	Timestamp ISO8601	定时任务的首次触发时间，取值为北京时间（UTC+8），按照ISO8601标准，格式：YYYY-MM-DDThh:mm:ss+08:00。 示例值：2018-08-28T23:00:00+08:00
EndTime	否	Timestamp ISO8601	定时任务的结束时间，取值为北京时间（UTC+8），按照ISO8601标准，格式：YYYY-MM-DDThh:mm:ss+08:00。 此参数与Recurrence需要同时指定，到达结束时间之后，定时任务将不再生效。 示例值：2019-01-01T00:00:00+08:00
Recurrence	否	String	定时任务的重复方式。为标准 Cron 格式。定时任务中的 Recurrence 参数限制 为5个字段，由空格分开，结构为：分，小时，日期，月份，星期。此参数与EndTime需要同时指定。 示例值：0 14 * /1 **

3. 输出参数

参数名称	类型	描述
ScheduledActionId	String	定时任务ID 示例值: asst-chwbkq4c
RequestId	String	唯一请求 ID, 由服务端生成, 每次请求都会返回 (若请求因其他原因未能抵达服务端, 则该次请求不会获得 RequestId)。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建单次执行的定时任务

创建定时任务, 在指定时间 (北京时间2018年8月28日23点) 调整伸缩组的最大实例数、最小实例数和期望实例数至 10、4、6。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateScheduledAction
<公共请求参数>

{
  "DesiredCapacity": 6,
  "AutoScalingGroupId": "asg-2nr9xh8h",
  "MinSize": 4,
  "MaxSize": 10,
  "ScheduledActionName": "scheduled-action-0",
  "StartTime": "2018-08-28T23:00:00+08:00"
}
```

输出示例

```
{
  "Response": {
    "ScheduledActionId": "asst-chwbkq4c",
    "RequestId": "193a710f-8dbf-46aa-8b4a-195532244df8"
  }
}
```

示例2 创建重复执行的定时任务

创建定时任务, 从北京时间2018年8月28日开始, 每天 23:00 调整伸缩组的最大实例数、最小实例数和期望实例数至 7、2、3, 在北京时间2019年1月1日 00:00 后结束。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateScheduledAction
<公共请求参数>
```

```
{
  "Recurrence": "0 23 * * *",
  "DesiredCapacity": 3,
  "AutoScalingGroupId": "asg-2nr9xh8h",
  "MinSize": 2,
  "MaxSize": 7,
  "ScheduledActionName": "scheduled-action-1",
  "StartTime": "2018-08-28T23:00:00+08:00",
  "EndTime": "2019-01-01T00:00:00+08:00"
}
```

输出示例

```
{
  "Response": {
    "ScheduledActionId": "asst-1e3us530",
    "RequestId": "502fd6fa-44ff-4c79-b77e-ee20f72bddc0"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.CronExpressionIllegal	定时任务指定的Cron表达式无效。

错误码	描述
InvalidParameterValue.EndTimeBeforeStartTime	定时任务设置的结束时间在开始时间。
InvalidParameterValue.InvalidAutoScalingGroupID	伸缩组ID无效。
InvalidParameterValue.InvalidScheduledActionNameIncludeIllegalChar	定时任务名称包含无效字符。
InvalidParameterValue.ScheduledActionNameDuplicate	定时任务名称重复。
InvalidParameterValue.Size	伸缩组最大数量、最小数量、期望实例数取值不合法。
InvalidParameterValue.StartTimeBeforeCurrentTime	定时任务设置的开始时间在当前时间之前。
InvalidParameterValue.TimeFormat	时间格式错误。
InvalidParameterValue.TooLong	取值过多。
LimitExceeded.DesiredCapacityLimitExceeded	期望实例数超出限制。
LimitExceeded.MaxSizeLimitExceeded	最大实例数大于限制。
LimitExceeded.MinSizeLimitExceeded	最小实例数低于限制。
LimitExceeded.QuotaNotEnough	当前剩余配额不足。
LimitExceeded.ScheduledActionLimitExceeded	定时任务数量超过限制。
MissingParameter	缺少参数错误。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。

告警触发策略相关接口

修改告警触发策略

最近更新时间：2025-05-16 01:09:31

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ModifyScalingPolicy）用于修改告警触发策略。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyScalingPolicy。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingPolicyId	是	String	告警策略ID。可通过登录 控制台 或调用接口 DescribeScalingPolicies ，取返回信息中的 AutoScalingPolicyId 获取告警策略ID。 示例值：asp-iir70sxx
ScalingPolicyName	否	String	告警策略名称，在您账号中必须唯一。名称长度不能超过60，名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点。 示例值：cpu_policy_test
AdjustmentType	否	String	告警触发后，期望实例数修改方式，仅适用于简单策略。取值范围： <ul style="list-style-type: none">CHANGE_IN_CAPACITY：增加或减少若干期望实例数EXACT_CAPACITY：调整至指定期望实例数PERCENT_CHANGE_IN_CAPACITY：按百分比调整期望实例数 示例值：CHANGE_IN_CAPACITY
AdjustmentValue	否	Integer	告警触发后，期望实例数的调整值，仅适用于简单策略。 <ul style="list-style-type: none">当 AdjustmentType 为 CHANGE_IN_CAPACITY 时，AdjustmentValue 为正数表示告警触发后增加实例，为负数表示告警触发后减少实例当 AdjustmentType 为 EXACT_CAPACITY 时，AdjustmentValue 的值即为告警触发后新的期望实例数，需要大于或等于0当 AdjustmentType 为 PERCENT_CHANGE_IN_CAPACITY 时，AdjustmentValue 为正数表示告警触发后按百分比增加实例，为负数表示告警触发后按百分比减少实例，单位是：%。 示例值：1
Cooldown	否	Integer	冷却时间，仅适用于简单策略，单位为秒。 示例值：60
MetricAlarm	否	MetricAlarm	告警监控指标，仅适用于简单策略。

参数名称	必选	类型	描述
PredefinedMetricType	否	String	预定义监控项，仅适用于目标追踪策略。取值范围： <ul style="list-style-type: none"> ASG_AVG_CPU_UTILIZATION：平均CPU使用率 ASG_AVG_LAN_TRAFFIC_OUT：平均内网出带宽 ASG_AVG_LAN_TRAFFIC_IN：平均内网入带宽 ASG_AVG_WAN_TRAFFIC_OUT：平均外网出带宽 ASG_AVG_WAN_TRAFFIC_IN：平均外网入带宽 示例值：ASG_AVG_CPU_UTILIZATION
TargetValue	否	Integer	目标值，仅适用于目标追踪策略。 <ul style="list-style-type: none"> ASG_AVG_CPU_UTILIZATION：[1, 100)，单位：% ASG_AVG_LAN_TRAFFIC_OUT：>0，单位：Mbps ASG_AVG_LAN_TRAFFIC_IN：>0，单位：Mbps ASG_AVG_WAN_TRAFFIC_OUT：>0，单位：Mbps ASG_AVG_WAN_TRAFFIC_IN：>0，单位：Mbps 示例值：10
EstimatedInstanceWarmup	否	Integer	实例预热时间，单位为秒，仅适用于目标追踪策略。取值范围为0-3600。 示例值：300
DisableScaleIn	否	Boolean	是否禁用缩容，仅适用于目标追踪策略。取值范围： <ul style="list-style-type: none"> true：目标追踪策略仅触发扩容 false：目标追踪策略触发扩容和缩容 示例值：false
NotificationUserGroupIds.N	否	Array of String	此参数已不再生效，请使用 创建通知 。 通知组ID，即为用户组ID集合。 示例值：["1678"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改简单类型告警触发策略

指定策略ID，修改简单策略相关属性。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyScalingPolicy
<公共请求参数>

{
  "AutoScalingPolicyId": "asp-iir70sxv",
  "Cooldown": 60,
  "ScalingPolicyName": "cpu_policy_test",
  "AdjustmentType": "CHANGE_IN_CAPACITY",
  "MetricAlarm": {
```

```
"Period": 60,
"ContinuousTime": 5,
"ComparisonOperator": "GREATER_THAN",
"Statistic": "AVERAGE",
"Threshold": 50,
"MetricName": "CPU_UTILIZATION"
},
"AdjustmentValue": "1"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "91413a64-9587-486b-ae4-9aba5e8a0068"
  }
}
```

示例2 修改目标追踪类型告警触发策略

指定策略ID，修改目标追踪策略相关属性。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyScalingPolicy
<公共请求参数>

{
  "AutoScalingPolicyId": "asp-9uroe7ta",
  "ScalingPolicyName": "target-tracking-test-policy",
  "DisableScaleIn": false,
  "EstimatedInstanceWarmup": 300,
  "TargetValue": 60,
  "PredefinedMetricType": "ASG_AVG_CPU_UTILIZATION"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "881a6752-55e8-4485-878a-a80065732a9f"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingPolicyId	告警策略ID无效。
InvalidParameterValue.InvalidNotificationUserGroupId	指定的通知组 ID 不是数值字符串格式。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.ScalingPolicyNameDuplicate	告警策略名称重复。
InvalidParameterValue.ThresholdOutOfRange	指定的阈值不在有效范围。
InvalidParameterValue.UserGroupIdNotFound	用户组不存在。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.ScalingPolicyNotFound	告警策略不存在。

查询告警触发策略

最近更新时间：2025-05-16 01:09:31

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeScalingPolicies）用于查询告警触发策略。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeScalingPolicies。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingPolicyIds.N	否	Array of String	按照一个或者多个告警策略ID查询。告警策略ID形如：asp-i9vkg894。每次请求的实例的上限为100。参数不支持同时指定AutoScalingPolicyIds和Filters。 示例值：["asp-5zffv598"]
Filters.N	否	Array of Filter	过滤条件。 <ul style="list-style-type: none">auto-scaling-policy-id - String - 是否必填：否 - （过滤条件）按照告警策略ID过滤。auto-scaling-group-id - String - 是否必填：否 - （过滤条件）按照伸缩组ID过滤。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups，取返回信息中的AutoScalingGroupId 获取伸缩组ID。scaling-policy-name - String - 是否必填：否 - （过滤条件）按照告警策略名称过滤。scaling-policy-type - String - 是否必填：否 - （过滤条件）按照告警策略类型过滤，取值范围为SIMPLE，TARGET_TRACKING。两者分别表示简单策略和目标追踪策略。 每次请求的Filters的上限为10，Filter.Values的上限为5。参数不支持同时指定AutoScalingPolicyIds和Filters。 示例值：[{"Name": "auto-scaling-group-id", "Values": ["asg-ewdpp04x"]}]]
Limit	否	Integer	返回数量，默认为20，最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值：20
Offset	否	Integer	偏移量，默认为0。关于Offset的更进一步介绍请参考 API 简介 中的相关小节。 示例值：0

3. 输出参数

参数名称	类型	描述
ScalingPolicySet	Array of ScalingPolicy	弹性伸缩告警触发策略详细信息列表。
TotalCount	Integer	符合条件的通知数量。

参数名称	类型	描述
		示例值: 1
RequestId	String	唯一请求 ID, 由服务端生成, 每次请求都会返回 (若请求因其他原因未能抵达服务端, 则该次请求不会获得 RequestId)。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询简单策略

指定策略ID查询简单类型告警触发策略

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeScalingPolicies
<公共请求参数>

{
  "AutoScalingPolicyIds": [
    "asp-7mme2ule"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "ScalingPolicySet": [
      {
        "EstimatedInstanceWarmup": null,
        "MetricAlarms": null,
        "AutoScalingGroupId": "asg-9dn1a5y6",
        "PredefinedMetricType": null,
        "ScalingPolicyType": "SIMPLE",
        "DisableScaleIn": null,
        "AutoScalingPolicyId": "asp-7mme2ule",
        "NotificationUserGroupIds": [],
        "Cooldown": 666,
        "ScalingPolicyName": "simple_policy_test",
        "AdjustmentType": "CHANGE_IN_CAPACITY",
        "MetricAlarm": {
          "ComparisonOperator": "GREATER_THAN",
          "Period": 60,
          "ContinuousTime": 5,
          "Threshold": 20,
          "Statistic": "AVERAGE",
          "PreciseThreshold": 20,
          "MetricName": "CPU_UTILIZATION"
        },
        "TargetValue": null,
      }
    ]
  }
}
```

```
"AdjustmentValue": 3
}
],
"RequestId": "297c6ed3-aa1c-43f4-be0f-10e513a86e6e"
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.CallMonitorError	Monitor接口调用失败。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidAutoScalingPolicyId	告警策略ID无效。
InvalidParameterValue.InvalidFilter	过滤条件无效。
LimitExceeded	超过配额限制。

删除告警触发策略

最近更新时间：2025-05-16 01:09:32

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DeleteScalingPolicy）用于删除告警触发策略。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteScalingPolicy。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingPolicyId	是	String	待删除的告警策略ID。可通过登录 控制台 或调用接口 DescribeScalingPolicies ，取返回信息中的 AutoScalingPolicyId 获取告警策略ID。 示例值：asp-lnfqbsq

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除告警策略

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteScalingPolicy
<公共请求参数>

{
  "AutoScalingPolicyId": "asp-lnfqbsq"
}
```


输出示例

```
{
  "Response": {
    "RequestId": "56e8b8fc-6043-4b3e-87b6-1a0be1c58f01"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingPolicyId	告警策略ID无效。
ResourceNotFound.ScalingPolicyNotFound	告警策略不存在。

创建告警触发策略

最近更新时间：2025-05-16 01:09:32

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（CreateScalingPolicy）用于创建告警触发策略。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateScalingPolicy。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-12wjuh0s
ScalingPolicyName	是	String	告警策略名称，在您账号中必须唯一。名称长度不能超过60，名称仅支持中文、英文、数字、下划线、分隔符“-”、小数点。 示例值：cpu_policy_test
ScalingPolicyType	否	String	告警触发策略类型，默认类型为SIMPLE。取值范围： <ul style="list-style-type: none">SIMPLE：简单策略TARGET_TRACKING：目标追踪策略 示例值：SIMPLE
AdjustmentType	否	String	告警触发后，期望实例数修改方式，仅适用于简单策略，在简单策略场景下必填。取值范围： <ul style="list-style-type: none">CHANGE_IN_CAPACITY：增加或减少若干期望实例数EXACT_CAPACITY：调整至指定期望实例数PERCENT_CHANGE_IN_CAPACITY：按百分比调整期望实例数 示例值：CHANGE_IN_CAPACITY
AdjustmentValue	否	Integer	告警触发后，期望实例数的调整值，仅适用于简单策略，在简单策略场景下必填。 <ul style="list-style-type: none">当 AdjustmentType 为 CHANGE_IN_CAPACITY 时，AdjustmentValue 为正数表示告警触发后增加实例，为负数表示告警触发后减少实例当 AdjustmentType 为 EXACT_CAPACITY 时，AdjustmentValue 的值即为告警触发后新的期望实例数，需要大于或等于0当 AdjustmentType 为 PERCENT_CHANGE_IN_CAPACITY 时，AdjustmentValue 为正数表示告警触发后按百分比增加实例，为负数表示告警触发后按百分比减少实例，单位是：%。

参数名称	必选	类型	描述
			示例值: 1
Cooldown	否	Integer	冷却时间, 单位为秒, 仅适用于简单策略。默认冷却时间300秒。 示例值: 60
MetricAlarm	否	MetricAlarm	告警监控指标, 仅适用于简单策略, 在简单策略场景下必填。 示例值: {"ComparisonOperator": "GREATER_THAN", "ContinuousTime": 1, "MetricName": "CPU_UTILIZATION", "Period": 60, "Statistic": "AVERAGE", "Threshold": 15}
PredefinedMetricType	否	String	预定义监控项, 仅适用于目标追踪策略, 在目标追踪策略场景下必填。取值范围: <ul style="list-style-type: none"> ASG_AVG_CPU_UTILIZATION: 平均CPU使用率 ASG_AVG_LAN_TRAFFIC_OUT: 平均内网出带宽 ASG_AVG_LAN_TRAFFIC_IN: 平均内网入带宽 ASG_AVG_WAN_TRAFFIC_OUT: 平均外网出带宽 ASG_AVG_WAN_TRAFFIC_IN: 平均外网入带宽 示例值: ASG_AVG_CPU_UTILIZATION
TargetValue	否	Integer	目标值, 仅适用于目标追踪策略, 在目标追踪策略场景下必填。 <ul style="list-style-type: none"> ASG_AVG_CPU_UTILIZATION: [1, 100), 单位: % ASG_AVG_LAN_TRAFFIC_OUT: >0, 单位: Mbps ASG_AVG_LAN_TRAFFIC_IN: >0, 单位: Mbps ASG_AVG_WAN_TRAFFIC_OUT: >0, 单位: Mbps ASG_AVG_WAN_TRAFFIC_IN: >0, 单位: Mbps 示例值: 10
EstimatedInstanceWarmup	否	Integer	实例预热时间, 单位为秒, 仅适用于目标追踪策略。取值范围为0-3600, 默认预热时间300秒。 示例值: 300
DisableScaleIn	否	Boolean	是否禁用缩容, 仅适用于目标追踪策略, 默认值为 false。取值范围: <ul style="list-style-type: none"> true: 目标追踪策略仅触发扩容 false: 目标追踪策略触发扩容和缩容 示例值: false
NotificationUserGroupIds.N	否	Array of String	此参数已不再生效, 请使用 创建通知 。 通知组ID, 即为用户组ID集合。 示例值: ["1678"]

3. 输出参数

参数名称	类型	描述
AutoScalingPolicyId	String	告警触发策略ID。 示例值: asp-iir70sxx
RequestId	String	唯一请求 ID, 由服务端生成, 每次请求都会返回 (若请求因其他原因未能抵达服务端, 则该次请求不会获得 RequestId)。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建简单策略: CPU 平均利用率

1分钟内, CPU平均利用率大于50%, 连续发生5次, 期望实例数增加1。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateScalingPolicy
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "Cooldown": 60,
  "ScalingPolicyType": "SIMPLE",
  "ScalingPolicyName": "cpu_policy_test",
  "AdjustmentType": "CHANGE_IN_CAPACITY",
  "MetricAlarm": {
    "Period": 60,
    "ContinuousTime": 5,
    "ComparisonOperator": "GREATER_THAN",
    "Statistic": "AVERAGE",
    "Threshold": 50,
    "MetricName": "CPU_UTILIZATION"
  },
  "AdjustmentValue": 1
}
```

输出示例

```
{
  "Response": {
    "AutoScalingPolicyId": "asp-iir70sxxv",
    "RequestId": "fb02c8bd-5f38-4786-91b6-0c6e06a88832"
  }
}
```

示例2 创建简单策略：内存平均利用率

1分钟内，内存平均利用率小于35%，连续发生5次，减少50%实例。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateScalingPolicy
<公共请求参数>

{
  "AutoScalingGroupId": "asg-12wjuh0s",
  "Cooldown": 300,
  "ScalingPolicyType": "SIMPLE",
  "ScalingPolicyName": "mem_policy_test",
  "AdjustmentType": "PERCENT_CHANGE_IN_CAPACITY",
  "MetricAlarm": {
    "Period": 60,
```

```
"ContinuousTime": 5,
"ComparisonOperator": "LESS_THAN",
"Statistic": "AVERAGE",
"Threshold": 50,
"MetricName": "MEM_UTILIZATION"
},
"AdjustmentValue": -50
}
```

输出示例

```
{
  "Response": {
    "AutoScalingPolicyId": "asp-f59pppuh",
    "RequestId": "116213d6-2269-44cc-af76-c4a8dc7dbdf9"
  }
}
```

示例3 创建目标追踪策略：CPU 平均利用率

保持伸缩组 CPU 平均利用率接近60%，告警触发的扩容实例预热300秒，关闭禁用缩容。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateScalingPolicy
<公共请求参数>

{
  "AutoScalingGroupId": "asg-9dn1a5y6",
  "ScalingPolicyName": "cpu_target_tracking_test",
  "ScalingPolicyType": "TARGET_TRACKING",
  "PredefinedMetricType": "ASG_AVG_CPU_UTILIZATION",
  "EstimatedInstanceWarmup": 300,
  "DisableScaleIn": false,
  "TargetValue": 60
}
```

输出示例

```
{
  "Response": {
    "AutoScalingPolicyId": "asp-ljr51ssq",
    "RequestId": "58a8ac17-e864-4bf6-81c7-c5dc63b0057d"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.CallMonitorError	Monitor接口调用失败。
InternalServerError.CallNotificationError	通知服务接口调用失败。
InternalServerError.RequestError	内部请求错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidNotificationUserGroupId	指定的通知组 ID 不是数值字符串格式。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.ScalingPolicyNameDuplicate	告警策略名称重复。
InvalidParameterValue.ThresholdOutOfRange	指定的阈值不在有效范围。
InvalidParameterValue.TooLong	取值过多。
InvalidParameterValue.UserGroupIdNotFound	用户组不存在。
LimitExceeded.QuotaNotEnough	当前剩余配额不足。
LimitExceeded.TargetTrackingScalingPolicy	每个伸缩组限制创建一个目标追踪策略。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。

触发伸缩策略

最近更新时间：2025-05-16 01:09:31

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ExecuteScalingPolicy）用于执行伸缩策略。

- 可以根据伸缩策略ID执行伸缩策略。
- 伸缩策略所属伸缩组处于伸缩活动时，会拒绝执行伸缩策略。
- 本接口不支持执行目标追踪策略。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ExecuteScalingPolicy。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingPolicyId	是	String	告警伸缩策略ID，不支持目标追踪策略。可通过 DescribeScalingPolicies 接口返回的 ScalingPolicyType 参数获取告警策略类型。 示例值：asp-f59pppuh
HonorCooldown	否	Boolean	是否检查伸缩组活动处于冷却时间内，默认值为false 示例值：false
TriggerSource	否	String	执行伸缩策略的触发来源，取值包括 API 和 CLOUD_MONITOR，默认值为 API。 CLOUD_MONITOR 专门供云监控触发调用。 示例值：API

3. 输出参数

参数名称	类型	描述
ActivityId	String	伸缩活动ID 示例值：asa-o4v87ae9
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 触发伸缩策略

触发告警伸缩策略

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ExecuteScalingPolicy
<公共请求参数>

{
  "HonorCooldown": "false",
  "AutoScalingPolicyId": "asp-f59pppuh"
}
```

输出示例

```
{
  "Response": {
    "ActivityId": "asa-o4v87ae9",
    "RequestId": "116213d6-2269-44cc-af76-c4a8dc7dbdf9"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.NoActivityToGenerate	未生成伸缩活动。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingPolicyId	告警策略ID无效。
InvalidParameterValue.TargetTrackingScalingPolicy	不支持执行目标追踪策略。
ResourceInUse.AutoScalingGroupNotActive	伸缩组处于禁用状态。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.ScalingPolicyNotFound	告警策略不存在。
ResourceUnavailable.AutoScalingGroupAbnormalStatus	伸缩组状态异常。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。

通知相关接口

修改通知

最近更新时间：2025-05-16 01:09:29

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（ModifyNotificationConfiguration）用于修改通知。

- 通知的接收端类型不支持修改。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyNotificationConfiguration。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingNotificationId	是	String	待修改的通知ID。可通过登录 控制台 或调用接口 DescribeNotificationConfigurations ，取返回信息中的 AutoScalingNotificationId 获取通知ID。 示例值：asn-2sestqbr
NotificationTypes.N	否	Array of String	通知类型，即为需要订阅的通知类型集合，取值范围如下： <ul style="list-style-type: none">SCALE_OUT_SUCCESSFUL：扩容成功SCALE_OUT_FAILED：扩容失败SCALE_IN_SUCCESSFUL：缩容成功SCALE_IN_FAILED：缩容失败REPLACE_UNHEALTHY_INSTANCE_SUCCESSFUL：替换不健康子机成功REPLACE_UNHEALTHY_INSTANCE_FAILED：替换不健康子机失败 示例值：["SCALE_IN_SUCCESSFUL","SCALE_IN_FAILED"]
NotificationUserGroupIds.N	否	Array of String	通知组ID，即为用户组ID集合，用户组ID可以通过 ListGroups 查询。该参数仅在 TargetType 为 USER_GROUP 时生效。 示例值：["1678"]
QueueName	否	String	TDMQ CMQ 队列名。 原CMQ已下线 ，目前仅推荐使用 TDMQ CMQ。该参数仅在 TargetType 为 TDMQ_CMQ_QUEUE 时生效。 示例值：queue
TopicName	否	String	TDMQ CMQ 主题名。 原CMQ已下线 ，目前仅推荐使用 TDMQ CMQ。该参数仅在 TargetType 为 TDMQ_CMQ_TOPIC 时生效。 示例值：topic

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改事件通知

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyNotificationConfiguration
<公共请求参数>

{
  "NotificationUserGroupIds": [
    "1678"
  ],
  "NotificationTypes": [
    "SCALE_IN_FAILED",
    "SCALE_IN_SUCCESSFUL"
  ],
  "AutoScalingNotificationId": "asn-2sestqbr"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "52e32a5b-5f69-4d48-a3f1-f2fea5c43a70"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.ConflictNotificationTarget	通知接收端类型冲突。
InvalidParameterValue.InvalidAutoScalingNotificationId	通知ID无效。
InvalidParameterValue.InvalidNotificationUserGroupId	指定的通知组 ID 不是数值字符串格式。
InvalidParameterValue.UserGroupIdNotFound	用户组不存在。
ResourceNotFound.AutoScalingNotificationNotFound	通知不存在。
ResourceNotFound.CmqQueueNotFound	指定的 CMQ queue 不存在。
ResourceNotFound.TDMQCMQQueueNotFound	TDMQ-CMQ 队列不存在。
ResourceNotFound.TDMQCMQTopicNotFound	TDMQ-CMQ 主题不存在。
ResourceUnavailable.TDMQCMQTopicHasNoSubscriber	TDMQ-CMQ 主题无订阅者。

查询通知

最近更新时间：2025-05-30 01:10:12

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口 (DescribeNotificationConfigurations) 用于查询一个或多个通知的详细信息。

可以根据通知ID、伸缩组ID等信息来查询通知的详细信息。过滤信息详细请见过滤器 `Filter`。

如果参数为空，返回当前用户一定数量（Limit所指定的数量，默认为20）的通知。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeNotificationConfigurations。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingNotificationIds.N	否	Array of String	按照一个或者多个通知ID查询，列表长度上限为100。可通过登录 控制台 获取通知ID。参数不支持同时指定AutoScalingNotificationIds和Filters。 示例值：["asn-9bhwxqhq"]
Filters.N	否	Array of Filter	过滤条件。 <ul style="list-style-type: none">auto-scaling-notification-id - String - 是否必填：否 - （过滤条件）按照通知ID过滤。auto-scaling-group-id - String - 是否必填：否 - （过滤条件）按照伸缩组ID过滤。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups，取返回信息中的AutoScalingGroupId 获取伸缩组ID。 每次请求的Filters的上限为10，Filter.Values的上限为5。参数不支持同时指定AutoScalingNotificationIds和Filters。 示例值：[{"Name": "auto-scaling-group-id", "Values": ["asg-hyfg539t"]}]
Limit	否	Integer	返回数量，默认为20，最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值：20
Offset	否	Integer	偏移量，默认为0。关于Offset的更进一步介绍请参考 API 简介 中的相关小节。 示例值：0

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的通知数量。 示例值：1

参数名称	类型	描述
AutoScalingNotificationSet	Array of AutoScalingNotification	弹性伸缩事件通知详细信息列表。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询通知

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeNotificationConfigurations
<公共请求参数>

{
  "AutoScalingNotificationIds": [
    "asn-9bhwxqgh"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "AutoScalingNotificationSet": [
      {
        "TargetType": "USER_GROUP",
        "TopicName": "topicname",
        "QueueName": "queuename",
        "AutoScalingGroupId": "asg-2umy3jbd",
        "NotificationUserGroupIds": [
          "1678"
        ],
        "NotificationTypes": [
          "SCALE_OUT_SUCCESSFUL"
        ],
        "AutoScalingNotificationId": "asn-9bhwxqgh"
      }
    ],
    "RequestId": "0539a5fc-14b8-4591-9fe2-faee32031a64"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterConflict	指定的两个参数冲突，不能同时存在。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidAutoScalingNotificationId	通知ID无效。
InvalidParameterValue.InvalidFilter	过滤条件无效。

删除通知

最近更新时间：2025-05-16 01:09:30

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DeleteNotificationConfiguration）用于删除特定的通知。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteNotificationConfiguration。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingNotificationId	否	String	待删除的通知ID，目前为必填参数。可通过登录 控制台 或调用接口 DescribeNotificationConfigurations ，取返回信息中的 AutoScalingNotificationId 获取通知ID。 示例值：asn-2sestqbr

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除通知

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteNotificationConfiguration
<公共请求参数>

{
  "AutoScalingNotificationId": "asn-2sestqbr"
}
```


输出示例

```
{
  "Response": {
    "RequestId": "12868c8c-b1a3-4b66-a03d-1a3cacadca1d"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidAutoScalingNotificationId	通知ID无效。
ResourceNotFound.AutoScalingNotificationNotFound	通知不存在。

创建通知

最近更新时间：2025-05-16 01:09:30

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（CreateNotificationConfiguration）用于创建通知。

通知到 CMQ 主题或队列时，消息内容如下：

```
{
  "Service": "Tencent Cloud Auto Scaling",
  "CreatedTime": "2021-10-11T10:15:11Z", // 活动创建时间
  "AppId": "100000000",
  "ActivityId": "asa-fznnvrja", // 伸缩活动ID
  "AutoScalingGroupId": "asg-pc2oqu2z", // 伸缩组ID
  "ActivityType": "SCALE_OUT", // 伸缩活动类型
  "StatusCode": "SUCCESSFUL", // 伸缩活动结果
  "Description": "Activity was launched in response to a difference between desired capacity and actual capacity, scale out 1 instance(s).", // 伸缩活动描述
  "StartTime": "2021-10-11T10:15:11Z", // 活动开始时间
  "EndTime": "2021-10-11T10:15:32Z", // 活动结束时间
  "DetailedStatusMessageSet": [ // 活动内部错误集合（非空不代表活动失败）
    {
      "Code": "InvalidInstanceType",
      "Zone": "ap-guangzhou-2",
      "InstanceId": "",
      "InstanceChargeType": "POSTPAID_BY_HOUR",
      "SubnetId": "subnet-4t5mgeuu",
      "Message": "The specified instance type `S5.LARGE8` is invalid in `subnet-4t5mgeuu`, `ap-guangzhou-2`.",
      "InstanceType": "S5.LARGE8"
    }
  ]
}
```

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateNotificationConfiguration。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

参数名称	必选	类型	描述
AutoScalingGroupId	是	String	伸缩组ID。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups ，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-12wjuh0s
NotificationTypes.N	是	Array of String	通知类型，即为需要订阅的通知类型集合，取值范围如下： <ul style="list-style-type: none"> SCALE_OUT_SUCCESSFUL：扩容成功 SCALE_OUT_FAILED：扩容失败 SCALE_IN_SUCCESSFUL：缩容成功 SCALE_IN_FAILED：缩容失败 REPLACE_UNHEALTHY_INSTANCE_SUCCESSFUL：替换不健康子机成功 REPLACE_UNHEALTHY_INSTANCE_FAILED：替换不健康子机失败 示例值：["SCALE_OUT_SUCCESSFUL","SCALE_OUT_FAILED"]
NotificationUserGroupIds.N	否	Array of String	通知组ID，即为用户组ID集合，用户组ID可以通过 ListGroups 查询。该参数仅在 TargetType 为 USER_GROUP 时生效。 示例值：["1678"]
TargetType	否	String	通知接收端类型，取值如下： <ul style="list-style-type: none"> USER_GROUP：用户组 TDMQ_CMQ_TOPIC：TDMQ CMQ 主题 TDMQ_CMQ_QUEUE：TDMQ CMQ 队列 CMQ_QUEUE：CMQ 队列，CMQ已下线，目前仅推荐使用 TDMQ CMQ。 CMQ_TOPIC：CMQ 主题，CMQ已下线，目前仅推荐使用 TDMQ CMQ。 默认值为：USER_GROUP。 示例值：USER_GROUP
QueueName	否	String	TDMQ CMQ 队列名，如 TargetType 取值为 TDMQ_CMQ_QUEUE，该字段必填。 示例值：queue
TopicName	否	String	TDMQ CMQ 主题名，如 TargetType 取值为 TDMQ_CMQ_TOPIC，该字段必填。 示例值：topic

3. 输出参数

参数名称	类型	描述
AutoScalingNotificationId	String	通知ID。 示例值：asn-2sestqbr
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建扩容成功、扩容失败事件通知

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateNotificationConfiguration
<公共请求参数>

{
```

```
"AutoScalingGroupId": "asg-12wjuh0s",
"NotificationUserGroupIds": [
  "1678"
],
"NotificationTypes": [
  "SCALE_OUT_FAILED",
  "SCALE_OUT_SUCCESSFUL"
]
}
```

输出示例

```
{
  "Response": {
    "AutoScalingNotificationId": "asn-2sestqbr",
    "RequestId": "fb02c8bd-5f38-4786-91b6-0c6e06a88832"
  }
}
```

示例2 创建扩容失败事件通知，接收端为 CMQ 队列

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateNotificationConfiguration
<公共请求参数>

{
  "AutoScalingGroupId": "asg-pc2oqu2z",
  "NotificationTypes": [
    "SCALE_OUT_FAILED"
  ],
  "TargetType": "CMQ_QUEUE",
  "QueueName": "test-queue"
}
```

输出示例

```
{
  "Response": {
    "AutoScalingNotificationId": "asn-03kyokwk",
    "RequestId": "8565bcc0-7b02-4e9c-ae9f-27e3d3d73e12"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.ConflictNotificationTarget	通知接收端类型冲突。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidNotificationUserGroupId	指定的通知组 ID 不是数值字符串格式。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.UserGroupIdNotFound	用户组不存在。
LimitExceeded	超过配额限制。
LimitExceeded.QuotaNotEnough	当前剩余配额不足。
MissingParameter	缺少参数错误。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.CmqQueueNotFound	指定的 CMQ queue 不存在。
ResourceNotFound.TDMQCMQQueueNotFound	TDMQ-CMQ 队列不存在。
ResourceNotFound.TDMQCMQTopicNotFound	TDMQ-CMQ 主题不存在。
ResourceUnavailable.CmqTopicHasNoSubscriber	指定的 CMQ Topic 无订阅者。
ResourceUnavailable.TDMQCMQTopicHasNoSubscriber	TDMQ-CMQ 主题无订阅者。

生命周期挂钩相关接口

升级生命周期挂钩

最近更新时间：2025-04-28 01:09:35

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（UpgradeLifecycleHook）用于升级生命周期挂钩。

- 本接口用于升级生命周期挂钩，采用“完全覆盖”风格，无论之前参数如何，统一按照接口参数设置为新的配置。对于非必填字段，不填写则按照默认值赋值。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：UpgradeLifecycleHook。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LifecycleHookId	是	String	生命周期挂钩ID。可以通过调用接口 DescribeLifecycleHooks ，取返回信息中的 LifecycleHookId 获取生命周期挂钩ID。 示例值：ash-8azjxzj9
LifecycleHookName	是	String	生命周期挂钩名称。名称仅支持中文、英文、数字、下划线（_）、短横线（-）、小数点（.），最大长度不能超128个字符。 示例值：hook-updated
LifecycleTransition	是	String	进行生命周期挂钩的场景，取值范围如下： * INSTANCE_LAUNCHING: 扩容生命周期挂钩 * INSTANCE_TERMINATING: 缩容生命周期挂钩 示例值：INSTANCE_LAUNCHING
DefaultResult	否	String	定义伸缩组在生命周期挂钩超时或 LifecycleCommand 执行失败时应采取的操作，取值范围是如下： * CONTINUE: 默认值，表示继续执行扩缩容活动 * ABANDON: 针对扩容挂钩，挂钩超时或 LifecycleCommand 执行失败的 CVM 实例会直接释放或移出；而针对缩容挂钩，会继续执行缩容活动。 示例值：CONTINUE
HeartbeatTimeout	否	Integer	生命周期挂钩超时之前可以经过的最长时间（以秒为单位），范围从30到7200秒，默认值为300秒 示例值：240
NotificationMetadata	否	String	弹性伸缩向通知目标发送的附加信息，配置通知时使用，默认值为空字符串。NotificationMetadata 和 LifecycleCommand 参数互斥，二者不可同时指定。 示例值：as-meta-0
NotificationTarget	否	NotificationTarget	通知目标。NotificationTarget和LifecycleCommand参数互斥，二者不可同时指定。

参数名称	必选	类型	描述
			示例值: {"QueueName": "as-pool", "TargetType": "TDMQ_CMQ_QUEUE"}
LifecycleTransitionType	否	String	进行生命周期挂钩的场景类型, 取值范围包括NORMAL 和 EXTENSION。说明: 设置为EXTENSION值, 在AttachInstances、DetachInstances、RemoveInstances接口时会触发生命周期挂钩操作, 值为NORMAL则不会在这些接口中触发生命周期挂钩。 示例值: NORMAL
LifecycleCommand	否	LifecycleCommand	远程命令执行对象。通知参数 NotificationMetadata、NotificationTarget 与 LifecycleCommand 互斥, 不可同时指定。 示例值: {"CommandId": "cmd-am4e3hd8", "Parameters": "{123: 222}"}

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID, 由服务端生成, 每次请求都会返回 (若请求因其他原因未能抵达服务端, 则该次请求不会获得 RequestId)。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 升级生命周期挂钩

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: UpgradeLifecycleHook
<公共请求参数>

{
  "DefaultResult": "CONTINUE",
  "LifecycleHookName": "hook-updated",
  "LifecycleHookId": "ash-8azjzxj9",
  "LifecycleTransition": "INSTANCE_LAUNCHING",
  "HeartbeatTimeout": "240"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "db656d36-ed6b-4795-bdc4-94e7a7e04acb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台, 方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallCmqError	Cmq 接口调用失败。
InternalServerError.CallStsError	STS 接口调用失败。
InternalServerError.CallTATErr	TAT 接口调用失败。
InvalidParameter	参数错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Filter	无效的过滤器。
InvalidParameterValue.InvalidLifecycleHookId	生命周期挂钩ID无效。
InvalidParameterValue.LifecycleHookNameDuplicated	生命周期挂钩名称重复。
MissingParameter	缺少参数错误。
ResourceNotFound.CommandNotFound	命令不存在。
ResourceNotFound.LifecycleHookNotFound	无法找到指定生命周期挂钩。
ResourceUnavailable.CmqTopicHasNoSubscriber	指定的 CMQ Topic 无订阅者。
ResourceUnavailable.TDMQCMQTopicHasNoSubscriber	TDMQ-CMQ 主题无订阅者。

查询生命周期挂钩

最近更新时间：2025-04-25 01:08:46

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeLifecycleHooks）用于查询生命周期挂钩信息。

- 可以根据伸缩组ID、生命周期挂钩ID或者生命周期挂钩名称等信息来查询生命周期挂钩的详细信息。过滤信息详细请见过滤器Filter。
- 如果参数为空，返回当前用户一定数量（Limit所指定的数量，默认为20）的生命周期挂钩。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeLifecycleHooks。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LifecycleHookIds.N	否	Array of String	按照一个或者多个生命周期挂钩ID查询。生命周期挂钩ID形如：ash-8azjzxc1。每次请求的上限为100。参数不支持同时指定LifecycleHookIds和Filters。 示例值：["ash-mzdlwrbt"]，
Filters.N	否	Array of Filter	过滤条件。 <ul style="list-style-type: none">lifecycle-hook-id - String - 是否必填：否 - （过滤条件）按照生命周期挂钩ID过滤。lifecycle-hook-name - String - 是否必填：否 - （过滤条件）按照生命周期挂钩名称过滤。auto-scaling-group-id - String - 是否必填：否 - （过滤条件）按照伸缩组ID过滤。 每次请求的Filters的上限为10，Filter.Values的上限为5。参数不支持同时指定LifecycleHookIds和Filters。 示例值：[{"Name": "auto-scaling-group-id", "Values": ["asg-0xrkdurh"]}]
Limit	否	Integer	返回数量，默认为20，最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值：20
Offset	否	Integer	偏移量，默认为0。关于Offset的更进一步介绍请参考 API 简介 中的相关小节。 示例值：0

3. 输出参数

参数名称	类型	描述
LifecycleHookSet	Array of LifecycleHook	生命周期挂钩数组
TotalCount	Integer	总体数量 示例值：4

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询生命周期挂钩

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeLifecycleHooks
<公共请求参数>

{}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "LifecycleHookSet": [
      {
        "LifecycleHookName": "terminate-topic",
        "LifecycleTransitionType": "NORMAL",
        "AutoScalingGroupId": "asg-8fbozqja",
        "HeartbeatTimeout": 120,
        "NotificationMetadata": "topic",
        "NotificationTarget": {
          "TargetType": "TDMQ_TOPIC",
          "TopicName": "one-topic",
          "QueueName": "as-tdmq-queue"
        },
        "LifecycleCommand": {
          "CommandId": "cmd-hyg5r4e3",
          "Parameters": "{123: 222}"
        },
        "CreatedTime": "2019-04-19T02:59:30Z",
        "DefaultResult": "ABANDON",
        "LifecycleHookId": "ash-oq76wsrx",
        "LifecycleTransition": "INSTANCE_TERMINATING"
      }
    ],
    "RequestId": "dff07f6e-bdbc-4532-baeb-e7fb3aeebe248"
  }
}
```

示例2 查询生命周期挂钩，使用Filter

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeLifecycleHooks
```

<公共请求参数>

```
{
  "Filters": [
    {
      "Values": [
        "asg-8fbozqja"
      ],
      "Name": "auto-scaling-group-id"
    },
    {
      "Values": [
        "ash-fbjiexz7"
      ],
      "Name": "lifecycle-hook-id"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "LifecycleHookSet": [
      {
        "LifecycleHookName": "launch-queue",
        "LifecycleTransitionType": "NORMAL",
        "AutoScalingGroupId": "asg-8fbozqja",
        "HeartbeatTimeout": 120,
        "NotificationMetadata": "queue",
        "NotificationTarget": {
          "TargetType": "TDMQ_QUEUE",
          "TopicName": "as-topic-0",
          "QueueName": "one-queue"
        },
        "LifecycleCommand": {
          "CommandId": "cmd-j87y5tr4",
          "Parameters": "{123: 222}"
        },
        "CreatedTime": "2019-04-19T02:57:14Z",
        "DefaultResult": "CONTINUE",
        "LifecycleHookId": "ash-fbjiexz7",
        "LifecycleTransition": "INSTANCE_LAUNCHING"
      }
    ],
    "RequestId": "2d774a6c-bcaa-4805-b0cd-bd64519e2538"
  }
}
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidFilter	过滤条件无效。
InvalidParameterValue.InvalidLifecycleHookId	生命周期挂钩ID无效。
MissingParameter	缺少参数错误。

删除生命周期挂钩

最近更新时间：2025-05-14 01:09:26

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DeleteLifecycleHook）用于删除生命周期挂钩。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteLifecycleHook。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LifecycleHookId	是	String	生命周期挂钩ID。可以通过调用接口 DescribeLifecycleHooks ，取返回信息中的 LifecycleHookId 获取生命周期挂钩ID。 示例值：ash-5rynk2n5

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除指定的生命周期挂钩

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteLifecycleHook
<公共请求参数>

{
  "LifecycleHookId": "ash-5rynk2n5"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "4351a2d1-5cf1-43cc-97d3-16444c86b29f"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue.InvalidLifecycleHookId	生命周期挂钩ID无效。
MissingParameter	缺少参数错误。
ResourceNotFound.LifecycleHookNotFound	无法找到指定生命周期挂钩。

创建生命周期挂钩

最近更新时间：2025-05-16 01:09:30

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（CreateLifecycleHook）用于创建生命周期挂钩。

- 您可以为生命周期挂钩配置消息通知或执行自动化助手命令。

如果您配置了通知消息，弹性伸缩会通知您的TDMQ消息队列，通知内容形如：

```
{
  "Service": "Tencent Cloud Auto Scaling",
  "Time": "2019-03-14T10:15:11Z",
  "AppId": "1251783334",
  "ActivityId": "asa-fznnvrja",
  "AutoScalingGroupId": "asg-ft6y7u8n",
  "LifecycleHookId": "ash-p9i7y6t5",
  "LifecycleHookName": "my-hook",
  "LifecycleActionToken": "3080e1c9-0efe-4dd7-ad3b-90cd6618298f",
  "InstanceId": "ins-y6dr5e43",
  "LifecycleTransition": "INSTANCE_LAUNCHING",
  "NotificationMetadata": ""
}
```

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateLifecycleHook。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-8fbozqja
LifecycleHookName	是	String	生命周期挂钩名称。名称仅支持中文、英文、数字、下划线（_）、短横线（-）、小数点（.），最大长度不能超128个字节。示例值：one-hook-default

参数名称	必选	类型	描述
LifecycleTransition	是	String	进行生命周期挂钩的场景，取值范围如下： <ul style="list-style-type: none"> INSTANCE_LAUNCHING: 扩容生命周期挂钩 INSTANCE_TERMINATING: 缩容生命周期挂钩 示例值：INSTANCE_LAUNCHING
DefaultResult	否	String	定义伸缩组在生命周期挂钩超时或 LifecycleCommand 执行失败时应采取的操作，取值范围如下： <ul style="list-style-type: none"> CONTINUE: 默认值，表示继续执行扩缩容活动 ABANDON: 针对扩容挂钩，挂钩超时或 LifecycleCommand 执行失败的 CVM 实例会直接释放或移出；而针对缩容挂钩，会继续执行缩容活动。 示例值：CONTINUE
HeartbeatTimeout	否	Integer	生命周期挂钩超时之前可以经过的最长时间（以秒为单位），范围从30到7200秒，默认值为300秒 示例值：300
NotificationMetadata	否	String	弹性伸缩向通知目标发送的附加信息，配置通知时使用，默认值为空字符串，最长不超过 1024 字符。NotificationMetadata 和 LifecycleCommand 参数互斥，二者不可同时指定。 示例值：additional-info
NotificationTarget	否	NotificationTarget	通知目标。NotificationTarget和LifecycleCommand参数互斥，二者不可同时指定。 示例值：{"QueueName": "as-pool", "TargetType": "TDMQ_CMQ_QUEUE"}
LifecycleTransitionType	否	String	进行生命周期挂钩的场景类型，取值范围包括 NORMAL 和 EXTENSION，默认值为 NORMAL。 说明：设置为EXTENSION值，在 AttachInstances 、 DetachInstances 、 RemoveInstances 、 StopAutoScalingInstances 、 StartAutoScalingInstances 、 StartInstanceRefresh 接口时会触发生命周期挂钩操作，值为NORMAL则不会在这些接口中触发生命周期挂钩。 示例值：NORMAL
LifecycleCommand	否	LifecycleCommand	远程命令执行对象。通知相关参数（NotificationTarget、NotificationMetadata）与该参数互斥，二者不可同时指定。 示例值：{"CommandId": "cmd-am4e3hd8", "Parameters": "{123: 222}"}

3. 输出参数

参数名称	类型	描述
LifecycleHookId	String	生命周期挂钩ID 示例值：ash-8azjzj9
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建生命周期挂钩

创建生命周期挂钩，在实例创建场景下生效，DefaultResult设置为ABANDON，HeartbeatTimeout设置为360秒。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
```



```
Content-Type: application/json
X-TC-Action: CreateLifecycleHook
<公共请求参数>

{
  "AutoScalingGroupId": "asg-8fbozqja",
  "DefaultResult": "ABANDON",
  "LifecycleHookName": "one-hook",
  "LifecycleTransition": "INSTANCE_LAUNCHING",
  "HeartbeatTimeout": 360
}
```

输出示例

```
{
  "Response": {
    "LifecycleHookId": "ash-heyubibl",
    "RequestId": "5e414011-3359-45bd-8ba4-5b503d3fd3f6"
  }
}
```

示例2 创建生命周期挂钩，采用默认值

创建生命周期挂钩，在实例创建场景下生效。DefaultResult采用默认值，即CONTINUE。HeartbeatTimeout采用默认值，即300秒。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateLifecycleHook
<公共请求参数>

{
  "AutoScalingGroupId": "asg-8fbozqja",
  "LifecycleHookName": "one-hook-default",
  "LifecycleTransition": "INSTANCE_LAUNCHING"
}
```

输出示例

```
{
  "Response": {
    "LifecycleHookId": "ash-8azjzj9",
    "RequestId": "4fa9fd2e-5b6c-49fe-9ba7-ed2ee62d8271"
  }
}
```

示例3 创建生命周期挂钩，通知TDMQ主题模型

创建生命周期挂钩，在实例销毁场景下生效，DefaultResult设置为ABANDON，HeartbeatTimeout设置为120秒，通知名为“one-topic”的TDMQ主题模型。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateLifecycleHook
<公共请求参数>

{
  "HeartbeatTimeout": 120,
  "AutoScalingGroupId": "asg-8fbozqja",
  "LifecycleHookName": "terminate-topic",
  "NotificationMetadata": "topic",
  "NotificationTarget": {
    "TargetType": "TDMQ_TOPIC",
    "TopicName": "one-topic"
  },
  "DefaultResult": "ABANDON",
  "LifecycleTransition": "INSTANCE_TERMINATING"
}
```

输出示例

```
{
  "Response": {
    "LifecycleHookId": "ash-oq76wsrx",
    "RequestId": "cdb7670b-0412-444f-9d2f-0da9cd07c410"
  }
}
```

示例4 创建生命周期挂钩，通知TDMQ队列模型

创建生命周期挂钩，在实例创建场景下生效，DefaultResult设置为CONTINUE，HeartbeatTimeout设置为120秒，通知名为“one-queue”的TDMQ队列模型。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateLifecycleHook
<公共请求参数>

{
  "HeartbeatTimeout": 120,
  "AutoScalingGroupId": "asg-8fbozqja",
  "LifecycleHookName": "launch-queue",
  "NotificationMetadata": "queue",
  "NotificationTarget": {
    "TargetType": "TDMQ_QUEUE",
    "QueueName": "one-queue"
  },
  "DefaultResult": "CONTINUE",
}
```

```
"LifecycleTransition": "INSTANCE_LAUNCHING"
}
```

输出示例

```
{
  "Response": {
    "LifecycleHookId": "ash-fbjiexz7",
    "RequestId": "d3cf27b7-3090-4317-9107-d2eac986a446"
  }
}
```

示例5 创建执行自动化助手命令的生命周期挂钩

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateLifecycleHook
<公共请求参数>

{
  "HeartbeatTimeout": 300,
  "AutoScalingGroupId": "asg-mp5ydedh",
  "LifecycleCommand": {
    "CommandId": "cmd-id9u9191",
    "Parameters": "{\"var1\": \"ck\"}"
  },
  "LifecycleHookName": "demo2",
  "DefaultResult": "CONTINUE",
  "LifecycleTransition": "INSTANCE_LAUNCHING"
}
```

输出示例

```
{
  "Response": {
    "LifecycleHookId": "ash-kjurql2y",
    "RequestId": "08f7bea5-3e0a-4280-9970-5d959a922b0b"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallCmqError	Cmq 接口调用失败。
InternalServerError.CallTATError	TAT 接口调用失败。
InternalServerError.CalleeError	调用其他服务异常。
InvalidParameter	参数错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Filter	无效的过滤器。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.LifecycleHookNameDuplicated	生命周期挂钩名称重复。
InvalidParameterValue.Range	取值超出指定范围。
LimitExceeded.QuotaNotEnough	当前剩余配额不足。
MissingParameter	缺少参数错误。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.CmqQueueNotFound	指定的 CMQ queue 不存在。
ResourceNotFound.CommandNotFound	命令不存在。
ResourceNotFound.TDMQCMQQueueNotFound	TDMQ-CMQ 队列不存在。
ResourceNotFound.TDMQCMQTopicNotFound	TDMQ-CMQ 主题不存在。
ResourceUnavailable.CmqTopicHasNoSubscriber	指定的 CMQ Topic 无订阅者。
ResourceUnavailable.TDMQCMQTopicHasNoSubscriber	TDMQ-CMQ 主题无订阅者。

完成生命周期动作

最近更新时间：2025-05-16 01:09:30

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（CompleteLifecycleAction）用于完成生命周期动作。

- 用户通过调用本接口，指定一个具体的生命周期挂钩的结果（“CONTINUE”或者“ABANDON”）。如果一直不调用本接口，则生命周期挂钩会在超时时按照“DefaultResult”进行处理。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CompleteLifecycleAction。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LifecycleHookId	是	String	生命周期挂钩ID。可以通过调用接口 DescribeLifecycleHooks ，取返回信息中的 LifecycleHookId 获取生命周期挂钩ID。 示例值：ash-fbjexz7
LifecycleActionResult	是	String	生命周期动作的结果，取值范围如下： <ul style="list-style-type: none">CONTINUE: 默认值，表示继续执行扩缩容活动ABANDON: 针对扩容挂钩，挂钩超时或 LifecycleCommand 执行失败的 CVM 实例会直接释放或移出；而针对缩容挂钩，会继续执行缩容活动。 示例值：CONTINUE
InstanceId	否	String	实例ID，InstanceId 和 LifecycleActionToken 参数必须填写其中一个。可通过登录 控制台 或调用接口 DescribeInstances ，取返回信息中的 InstanceId 获取实例ID。 示例值：ins-ni8bpmve
LifecycleActionToken	否	String	生命周期动作令牌，InstanceId 和 LifecycleActionToken 必须填写其中一个。 该参数获取方式如下：配置 NotificationTarget 参数的挂钩被触发时，向 NotificationTarget 参数中指定的消息队列投递包含令牌的消息，消息队列的消费者可从消息中获取令牌。 示例值：34b59a82-6435-4516-90db-c98aeb46769b

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 完成生命周期动作，通过InstanceId调用

通过InstanceId参数，完成指定实例的生命周期挂钩

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CompleteLifecycleAction
<公共请求参数>

{
  "InstanceId": "ins-ni8bpmve",
  "LifecycleActionResult": "CONTINUE",
  "LifecycleHookId": "ash-fbjiexz7"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d0cf47b9-4236-491c-bfab-106947198afe"
  }
}
```

示例2 完成生命周期动作，通过LifecycleActionToken调用

通过LifecycleActionToken参数，完成指定实例的生命周期挂钩，生命周期挂钩发送给TDMQ的通知消息中包含LifecycleActionToken参数

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CompleteLifecycleAction
<公共请求参数>

{
  "LifecycleActionResult": "CONTINUE",
  "LifecycleHookId": "ash-fbjiexz7",
  "LifecycleActionToken": "4d910016-2590-444d-8f4a-c14940036902"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "de792ffe-e37e-4f1d-8534-300b555739da"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.MustOneParameter	参数缺失，两种参数之中必须指定其中一个。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.InvalidLifecycleHookId	生命周期挂钩ID无效。
ResourceNotFound.LifecycleHookInstanceNotFound	生命周期挂钩对应实例不存在。
ResourceNotFound.LifecycleHookNotFound	无法找到指定生命周期挂钩。
ResourceNotFound.LifecycleHookTokenNotFound	指定的生命周期挂钩Token不存在。
ResourceUnavailable.LifecycleActionResultHasSet	生命周期动作已经被设置。

修改生命周期挂钩

最近更新时间：2025-04-28 01:09:35

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

此接口用于修改生命周期挂钩。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyLifecycleHook。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LifecycleHookId	是	String	生命周期挂钩ID。可以通过调用接口 DescribeLifecycleHooks ，取返回信息中的 LifecycleHookId 获取生命周期挂钩ID。 示例值：ash-je1esoo9
LifecycleHookName	否	String	生命周期挂钩名称。名称仅支持中文、英文、数字、下划线（_）、短横线（-）、小数点（.），最大长度不能超128。 示例值：hook-test
LifecycleTransition	否	String	进入生命周期挂钩场景，取值范围如下： * INSTANCE_LAUNCHING: 扩容生命周期挂钩 * INSTANCE_TERMINATING: 缩容生命周期挂钩 示例值：INSTANCE_LAUNCHING
DefaultResult	否	String	定义伸缩组在生命周期挂钩超时或 LifecycleCommand 执行失败时应采取的操作，取值范围如下： * CONTINUE: 默认值，表示继续执行扩缩容活动 * ABANDON: 针对扩容挂钩，挂钩超时或 LifecycleCommand 执行失败的 CVM 实例会直接释放或移出；而针对缩容挂钩，会继续执行缩容活动。 示例值：CONTINUE
HeartbeatTimeout	否	Integer	生命周期挂钩超时之前可以经过的最长时间（以秒为单位），范围从 30 到 7200 秒。 示例值：30
NotificationMetadata	否	String	弹性伸缩向通知目标发送的附加信息。NotificationMetadata 与 LifecycleCommand 互斥，二者不可同时指定。 示例值：addtional-info
LifecycleTransitionType	否	String	进行生命周期挂钩的场景类型，取值范围包括 NORMAL 和 EXTENSION。说明：设置为 EXTENSION 值，在 AttachInstances、DetachInstances、RemoveInstances 接口时会触发生命周期挂钩操作，值为 NORMAL 则不会在这些接口中触发生命周期挂钩。 示例值：NORMAL

参数名称	必选	类型	描述
NotificationTarget	否	NotificationTarget	通知目标信息。NotificationTarget 与 LifecycleCommand 互斥，二者不可同时指定。 示例值：{"QueueName": "as-pool", "TargetType": "TDMQ_CMQ_QUEUE"}
LifecycleCommand	否	LifecycleCommand	远程命令执行对象。通知参数 NotificationMetadata、NotificationTarget 与 LifecycleCommand 互斥，不可同时指定。 示例值：{"CommandId": "cmd-am4e3hd8", "Parameters": "{123: 222}"}

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改生命周期挂钩至实例启动后进入挂钩时间

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyLifecycleHook
<公共请求参数>

{
  "LifecycleHookId": "ash-je1esoo9",
  "LifecycleTransition": "INSTANCE_LAUNCHING"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "4942c041-fc7f-4f50-b489-d01cdeb6638f"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.CallTATError	TAT 接口调用失败。
InvalidParameter	参数错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Range	取值超出指定范围。
ResourceNotFound.CmqQueueNotFound	指定的 CMQ queue 不存在。
ResourceNotFound.CommandNotFound	命令不存在。
ResourceNotFound.LifecycleHookNotFound	无法找到指定生命周期挂钩。
ResourceNotFound.TDMQCMQQueueNotFound	TDMQ-CMQ 队列不存在。
ResourceNotFound.TDMQCMQTopicNotFound	TDMQ-CMQ 主题不存在。
ResourceUnavailable.TDMQCMQTopicHasNoSubscriber	TDMQ-CMQ 主题无订阅者。

实例刷新相关接口

暂停实例刷新

最近更新时间：2025-04-28 01:09:35

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

暂停正在执行的实例刷新活动。

- 暂停状态下，伸缩组也会处于停用中状态
- 当前正在更新或扩容的实例不会暂停，待更新的实例会暂停更新
- 进行缩容时，所有实例都已经更新完成，此时无法暂停

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：StopInstanceRefresh。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过以下方式获取可用的伸缩组ID： <ul style="list-style-type: none">• 通过登录 控制台 查询伸缩组ID。• 通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-u7dy6t5r
RefreshActivityId	是	String	刷新活动ID。可以通过调用接口 DescribeRefreshActivities ，取返回信息中的 RefreshActivityId 获取实例刷新活动ID。 示例值：asr-dhwt6r54

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 暂停实例刷新活动

暂停伸缩组 asg-h7tgd87d 的实例刷新活动 asr-juhf6ytr

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: StopInstanceRefresh
<公共请求参数>

{
  "AutoScalingGroupId": "asg-h7tgd87d",
  "RefreshActivityId": "asr-juh6ytr"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c4190090-bc60-4f48-b9d4-48095b9596db"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
ResourceNotFound.RefreshActivityNotFound	指定的实例刷新活动不存在。
ResourceUnavailable.RefreshActivityStatusConflictWithOperation	实例刷新活动状态与当前操作冲突。

错误码	描述
UnauthorizedOperation.AutoScalingRoleUnauthorized	未授权弹性伸缩预设服务角色AS_QCSRole，请前往弹性伸缩控制台进行授权。

启动实例刷新

最近更新时间：2025-06-11 01:05:01

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

根据启动配置中参数，刷新伸缩组内运行中状态 CVM 实例，返回实例刷新活动的 RefreshActivityId。

- 对于重装实例的刷新方式（目前仅支持重装），重装时仅会从启动配置中获取 ImageId、UserData、EnhancedService、HostName、LoginSettings 参数进行刷新，实例的其他参数不会刷新
- 实例刷新期间（包括暂停状态），伸缩组会被停用。不建议刷新期间修改关联启动配置，否则会影响刷新参数，造成实例配置不一致
- 滚动更新模式会分成多批次进行刷新实例，单批次中若存在刷新失败实例，活动会进入失败暂停状态
- 若待刷新实例被移出或销毁，会被标记为 NOT_FOUND 状态，不阻塞实例刷新活动
- 运行中状态实例与最新启动配置参数一致，实例也会再次刷新

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：StartInstanceRefresh。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。 示例值：asg-12yqet78
RefreshSettings	是	RefreshSettings	刷新设置。
RefreshMode	否	String	刷新模式。默认值为 ROLLING_UPDATE_RESET，取值范围： <ul style="list-style-type: none">ROLLING_UPDATE_RESET：重装系统进行滚动更新ROLLING_UPDATE_REPLACE：新建实例替换进行滚动更新，该模式暂不支持回滚接口 示例值：ROLLING_UPDATE_RESET

3. 输出参数

参数名称	类型	描述
RefreshActivityId	String	刷新活动 ID。 示例值：asr-6t7y8u5r
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 滚动更新方式进行实例刷新

将伸缩组 asg-9dn1a5y6 中的实例分成 3 个批次进行滚动更新，批次间会自动暂停。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: StartInstanceRefresh
<公共请求参数>

{
  "AutoScalingGroupId": "asg-9dn1a5y6",
  "RefreshMode": "ROLLING_UPDATE_RESET",
  "RefreshSettings": {
    "RollingUpdateSettings": {
      "BatchNumber": 3,
      "BatchPause": "BATCH_INTERVAL_PAUSE"
    }
  }
}
```

输出示例

```
{
  "Response": {
    "RefreshActivityId": "asr-y67t5r4e",
    "RequestId": "c4190090-bc60-4f48-b9d4-48095b9596db"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

• [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameterValue.BatchNumberTooLarge	批次数不能大于待刷新实例总数。
InvalidParameterValue.MaxSurgeTooLarge	最大额外数量取值过大。
ResourceNotFound.AutoScalingGroupIdNotFound	伸缩组不存在。
ResourceUnavailable.AutoScalingGroupAbnormalStatus	伸缩组状态异常。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.AutoScalingGroupInRefreshActivity	伸缩组处于另一个实例刷新活动当中。
ResourceUnavailable.InquiryPriceResetInstanceFailed	重装实例询价失败，新镜像与该实例其他参数存在冲突或新镜像不存在。
ResourceUnavailable.NoInstanceCanRefresh	伸缩组内不存在运行中状态的实例，无法进行实例刷新。
ResourceUnavailable.RefreshActivityStatusConflictWithOperation	实例刷新活动状态与当前操作冲突。
UnauthorizedOperation.AutoScalingRoleUnauthorized	未授权弹性伸缩预设服务角色AS_QCSRole，请前往弹性伸缩控制台进行授权。

回滚实例刷新

最近更新时间：2025-04-28 01:09:35

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

回滚操作会生成一个新的实例刷新活动，该活动也支持分批次刷新以及暂停、恢复、取消操作，接口返回回滚活动的 RefreshActivityId。

- 原活动中待刷新实例变更为已取消，忽略不存在实例，其他状态实例进入回滚流程
- 原活动中正在刷新的实例不会立刻终止，刷新结束后再执行回滚活动
- 暂停状态或最近一次成功的刷新活动支持回滚，其他状态不支持回滚
- 原活动刷新方式为重装实例时，对于 ImageId 参数，会自动恢复到回滚前镜像 ID；对于 UserData、EnhancedService、LoginSettings、HostName 参数，依然会从启动配置中读取，需用户在回滚前自行修改启动配置
- 回滚活动暂不支持 MaxSurge 参数

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：RollbackInstanceRefresh。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过以下方式获取可用的伸缩组ID： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-12yqet78
RefreshSettings	是	RefreshSettings	刷新设置。 示例值：{"RollingUpdateSettings":{"BatchNumber": 1, "BatchPause": "AUTOMATIC"}, "CheckInstanceTargetHealth": true}
OriginRefreshActivityId	是	String	原始刷新活动 ID。可以通过调用接口 DescribeRefreshActivities ，取返回信息中的 OriginRefreshActivityId 获取原始刷新活动ID。 示例值：asr-fdsf8u6b
RefreshMode	否	String	刷新模式，目前仅支持滚动更新，默认值为 ROLLING_UPDATE_RESET。 示例值：ROLLING_UPDATE_RESET

3. 输出参数

参数名称	类型	描述
RefreshActivityId	String	刷新活动 ID。 示例值：asr-6t7y8u5r

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 滚动更新方式回滚实例刷新活动

对伸缩组 asg-9dn1a5y6 的实例刷新活动 asr-i8u7tytd 进行回滚。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: StartInstanceRefresh
<公共请求参数>

{
  "AutoScalingGroupId": "asg-9dn1a5y6",
  "RefreshMode": "ROLLING_UPDATE_RESET",
  "OriginRefreshActivityId": "asr-i8u7tytd",
  "RefreshSettings": {
    "RollingUpdateSettings": {
      "BatchNumber": 3,
      "BatchPause": "BATCH_INTERVAL_PAUSE"
    }
  }
}
```

输出示例

```
{
  "Response": {
    "RefreshActivityId": "asr-y67t5r4e",
    "RequestId": "c4190090-bc60-4f48-b9d4-48095b9596db"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameterValue.BatchNumberTooLarge	批次数不能大于待刷新实例总数。
ResourceNotFound.AutoScalingGroupIdNotFound	伸缩组不存在。
ResourceUnavailable.AutoScalingGroupAbnormalStatus	伸缩组状态异常。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.AutoScalingGroupInRefreshActivity	伸缩组处于另一个实例刷新活动当中。
ResourceUnavailable.InquiryPriceResetInstanceFailed	重装实例询价失败，新镜像与该实例其他参数存在冲突或新镜像不存在。
ResourceUnavailable.NoInstanceCanRefresh	伸缩组内不存在运行中状态的实例，无法进行实例刷新。
ResourceUnavailable.NoInstanceCanRollback	伸缩组中不存在可回滚的实例。
ResourceUnavailable.RefreshActivityCanNotRollback	当前刷新活动为成功状态，且不是最近一次执行的刷新活动，无法回滚。
ResourceUnavailable.RefreshActivityStatusConflictWithOperation	实例刷新活动状态与当前操作冲突。
ResourceUnavailable.RollbackTypeActivityCanNotRollbackAgain	回滚类型的实例刷新活动无法再次回滚。
UnauthorizedOperation.AutoScalingRoleUnauthorized	未授权弹性伸缩预设服务角色AS_QCSRole，请前往弹性伸缩控制台进行授权。

恢复实例刷新

最近更新时间：2025-04-28 01:09:36

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

恢复暂停状态的实例刷新活动，使其重试当前批次刷新失败实例或继续刷新后续批次，非暂停状态下调用该接口无效。

- 使用 MaxSurge 参数时活动可能会处于扩容或缩容失败导致的暂停状态，也可以使用该接口重试扩缩容。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ResumeInstanceRefresh。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可通过如下方式获取： <ul style="list-style-type: none">通过登录 控制台 查询伸缩组ID。通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-u7dy6t5r
RefreshActivityId	是	String	刷新活动ID。可以通过调用接口 DescribeRefreshActivities ，取返回信息中的 RefreshActivityId 获取实例刷新活动ID。 示例值：asr-dhwt6r54
ResumeMode	否	String	当前批次刷新失败实例的恢复方式，如不存在失败实例，该参数无效。默认值为RETRY，取值范围如下： <ul style="list-style-type: none">RETRY：重试当前批次刷新失败实例CONTINUE：跳过当前批次刷新失败实例 示例值：RETRY

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 恢复实例刷新活动

恢复伸缩组 asg-h7tgd87d 的实例刷新活动 asr-juh6ytr，使其重试当前批次中刷新失败的实例。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ResumeInstanceRefresh
<公共请求参数>

{
  "AutoScalingGroupId": "asg-h7tgd87d",
  "RefreshActivityId": "asr-juhF6ytr",
  "ResumeMode": "RETRY"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c4190090-bc60-4f48-b9d4-48095b9596db"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
ResourceNotFound.RefreshActivityNotFound	指定的实例刷新活动不存在。

错误码	描述
ResourceUnavailable.RefreshActivityStatusConflictWithOperation	实例刷新活动状态与当前操作冲突。
UnauthorizedOperation.AutoScalingRoleUnauthorized	未授权弹性伸缩预设服务角色AS_QCSRole，请前往弹性伸缩控制台进行授权。

查询实例刷新活动

最近更新时间：2025-06-11 01:05:02

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeRefreshActivities）用于查询伸缩组的实例刷新活动记录。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeRefreshActivities。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
RefreshActivityIds.N	否	Array of String	刷新活动ID列表。ID形如：asr-512ejpfo。每次请求的上限为100。参数不支持同时指定 RefreshActivityIds和Filters。 示例值：["asr-aaaabbbb"]
Filters.N	否	Array of Filter	过滤条件。 <ul style="list-style-type: none">• auto-scaling-group-id - String - 是否必填：否 - （过滤条件）按照伸缩组ID过滤。可通过登录 控制台 或调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。• refresh-activity-status-code - String - 是否必填：否 - （过滤条件）按照刷新活动状态过滤。（INIT：初始化中 RUNNING：运行中 SUCCESSFUL：活动成功 FAILED_PAUSE：失败暂停 AUTO_PAUSE：自动暂停 MANUAL_PAUSE：手动暂停 CANCELLED：活动取消 FAILED：活动失败）• refresh-activity-type - String - 是否必填：否 - （过滤条件）按照刷新活动类型过滤。（NORMAL：正常刷新活动 ROLLBACK：回滚刷新活动）• refresh-activity-id - String - 是否必填：否 - （过滤条件）按照刷新活动ID过滤。 每次请求的Filters的上限为10，Filter.Values的上限为5。参数不支持同时指定 RefreshActivityIds和Filters。 示例值：[{"Name": "auto-scaling-group-id", "Values": ["asg-ewdpp04x"]}]
Limit	否	Integer	返回数量，默认为20，最大值为100。关于Limit的更进一步介绍请参考 API 简介 中的相关小节。 示例值：100
Offset	否	Integer	偏移量，默认为0。关于Offset的更进一步介绍请参考 API 简介 中的相关小节。 示例值：0

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的刷新活动数量。

参数名称	类型	描述
		示例值: 1
RefreshActivitySet	Array of RefreshActivity	符合条件的刷新活动信息集合。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 使用Filters查看刷新活动列表

使用Filters查询刷新活动 asr-cs0fxpcu。

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeRefreshActivities
<公共请求参数>

{
  "Filters": [
    {
      "Name": "refresh-activity-id",
      "Values": [
        "asr-cs0fxpcu"
      ]
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "RefreshActivitySet": [
      {
        "Status": "SUCCESSFUL",
        "RefreshBatchSet": [
          {
            "RefreshBatchNum": 3,
            "EndTime": "2023-09-07T16:32:56Z",
            "RefreshBatchRelatedInstanceSet": [
              {
                "InstanceId": "ins-w188n531",
                "InstanceStatusMessage": "success",
                "LastActivityId": "asa-pb2y1bna",
                "InstanceStatus": "SUCCESSFUL"
              }
            ]
          }
        ]
      }
    ]
  }
}
```



```
"StartTime": "2023-09-07T16:31:44Z",
"RefreshBatchStatus": "SUCCESSFUL"
},
{
  "RefreshBatchNum": 2,
  "EndTime": "2023-09-07T16:31:44Z",
  "RefreshBatchRelatedInstanceSet": [
    {
      "InstanceId": "ins-w1824wre",
      "InstanceStatusMessage": "success",
      "LastActivityId": "asa-e8dqz8r6",
      "InstanceStatus": "SUCCESSFUL"
    },
    {
      "InstanceId": "ins-w188n53m",
      "InstanceStatusMessage": "success",
      "LastActivityId": "asa-e8dqz8r6",
      "InstanceStatus": "SUCCESSFUL"
    }
  ],
  "StartTime": "2023-09-07T16:29:56Z",
  "RefreshBatchStatus": "SUCCESSFUL"
},
{
  "RefreshBatchNum": 1,
  "EndTime": "2023-09-07T16:29:55Z",
  "RefreshBatchRelatedInstanceSet": [
    {
      "InstanceId": "ins-w188n53k",
      "InstanceStatusMessage": "success",
      "LastActivityId": "asa-e0o3uxsu",
      "InstanceStatus": "SUCCESSFUL"
    },
    {
      "InstanceId": "ins-w188n53n",
      "InstanceStatusMessage": "success",
      "LastActivityId": "asa-e0o3uxsu",
      "InstanceStatus": "SUCCESSFUL"
    }
  ],
  "StartTime": "2023-09-07T16:27:49Z",
  "RefreshBatchStatus": "SUCCESSFUL"
}
],
"RefreshMode": "ROLLING_UPDATE_RESET",
"AutoScalingGroupId": "asg-kgujlegg",
"ActivityType": "NORMAL",
"RefreshSettings": {
  "CheckInstanceTargetHealth": true,
  "RollingUpdateSettings": {
    "BatchNumber": 3,
    "BatchPause": "AUTOMATIC"
  }
}
```

```
},
"OriginRefreshActivityId": "asr-jy6t4e3w",
"CurrentRefreshBatchNum": 3,
"RefreshActivityId": "asr-cs0fxpcu",
"StartTime": "2023-09-07T16:27:49Z",
"CreatedTime": "2023-09-07T16:27:49Z",
"EndTime": "2023-09-07T16:32:56Z"
}
],
"RequestId": "48869660-f1df-4085-a62c-1f9636cbe030"
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameterValue.Filter	无效的过滤器。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidFilter	过滤条件无效。
InvalidParameterValue.LimitExceeded	取值超出限制。
LimitExceeded.FilterValuesTooLong	特定过滤器的值过多。

取消实例刷新

最近更新时间：2025-04-28 01:09:36

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

取消伸缩组的实例刷新活动。

- 已刷新的批次不受影响，待刷新批次被取消
- 如存在正在刷新的批次，不允许取消；可先暂停活动，等待当前批次结束后再取消
- 刷新失败的实例保持备用中状态，需用户手动处理后尝试退出备用中状态或销毁
- 取消后不允许回滚操作，也不支持恢复操作
- 因 maxSurge 参数而临时扩容的实例在取消后会自动销毁
- 进行缩容时，所有实例都已经更新完成，此时无法取消

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CancelInstanceRefresh。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AutoScalingGroupId	是	String	伸缩组ID。可以通过如下方式获取可用的伸缩组ID： <ul style="list-style-type: none">• 通过登录 控制台 查询伸缩组ID。• 通过调用接口 DescribeAutoScalingGroups，取返回信息中的 AutoScalingGroupId 获取伸缩组ID。 示例值：asg-u7dy6t5r
RefreshActivityId	是	String	刷新活动ID。可以通过调用接口 DescribeRefreshActivities ，取返回信息中的 RefreshActivityId 获取实例刷新活动ID。 示例值：asr-dhwt6r54

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 取消实例刷新活动

取消伸缩组 asg-h7tgd87d 的实例刷新活动 asr-juhf6ytr

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: CancelInstanceRefresh
```

<公共请求参数>

```
{
  "AutoScalingGroupId": "asg-h7tgd87d",
  "RefreshActivityId": "asr-juh6ytr"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c4190090-bc60-4f48-b9d4-48095b9596db"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
ResourceNotFound.RefreshActivityNotFound	指定的实例刷新活动不存在。
ResourceUnavailable.RefreshActivityStatusConflictWithOperation	实例刷新活动状态与当前操作冲突。
UnauthorizedOperation.AutoScalingRoleUnauthorized	未授权弹性伸缩预设服务角色AS_QCSRole，请前往弹性伸缩控制台进行授权。

其他接口

查询用户账号的资源限制

最近更新时间：2025-04-25 01:08:50

1. 接口描述

接口请求域名：as.tencentcloudapi.com。

本接口（DescribeAccountLimits）用于查询用户账号在弹性伸缩中的资源限制。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeAccountLimits。
Version	是	String	公共参数 ，本接口取值：2018-04-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

3. 输出参数

参数名称	类型	描述
MaxNumberOfLaunchConfigurations	Integer	用户账号被允许创建的启动配置最大数量 示例值：20
NumberOfLaunchConfigurations	Integer	用户账号启动配置的当前数量 示例值：15
MaxNumberOfAutoScalingGroups	Integer	用户账号被允许创建的伸缩组最大数量 示例值：30
NumberOfAutoScalingGroups	Integer	用户账号伸缩组的当前数量 示例值：25
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询启动配置和伸缩组的限制

输入示例

```
POST / HTTP/1.1
Host: as.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: DescribeAccountLimits
```

```
<公共请求参数>
```

```
{}
```

输出示例

```
{
  "Response": {
    "NumberOfLaunchConfigurations": 15,
    "MaxNumberOfLaunchConfigurations": 20,
    "NumberOfAutoScalingGroups": 25,
    "MaxNumberOfAutoScalingGroups": 30,
    "RequestId": "0c243e3a-70e0-4365-98b1-5fe22b4498a1"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter.ActionNotFound	无效的Action请求。

数据结构

最近更新时间：2025-06-02 01:09:20

Activity

符合条件的伸缩活动相关信息。

被如下接口引用：DescribeAutoScalingActivities, DescribeAutoScalingGroupLastActivities。

名称	类型	描述
AutoScalingGroupId	String	伸缩组ID。 示例值：asg-jub6fr43
ActivityId	String	伸缩活动ID。 示例值：asa-jdh6r43e
ActivityType	String	伸缩活动类型。取值如下： <ul style="list-style-type: none">SCALE_OUT：扩容活动SCALE_IN：缩容活动ATTACH_INSTANCES：添加实例REMOVE_INSTANCES：销毁实例DETACH_INSTANCES：移出实例TERMINATE_INSTANCES_UNEXPECTEDLY：实例在CVM控制台被销毁REPLACE_UNHEALTHY_INSTANCE：替换不健康实例START_INSTANCES：开启实例STOP_INSTANCES：关闭实例INVOKE_COMMAND：执行命令 示例值：SCALE_OUT
StatusCode	String	伸缩活动状态。取值如下 <ul style="list-style-type: none">INIT：初始化中RUNNING：运行中SUCCESSFUL：活动成功PARTIALLY_SUCCESSFUL：活动部分成功FAILED：活动失败CANCELLED：活动取消 示例值：RUNNING
StatusMessage	String	伸缩活动状态描述。 示例值：Success
Cause	String	伸缩活动起因。 示例值：Activity was launched in response to a difference between desired capacity and actual capacity.
Description	String	伸缩活动描述。 示例值：Activity was launched in response to a difference between desired capacity and actual capacity, scale out 1 instance(s).
StartTime	Timestamp ISO8601	伸缩活动开始时间，为 UTC 标准时间。 示例值：2024-11-12T05:30:21Z
EndTime	Timestamp ISO8601	伸缩活动结束时间，为 UTC 标准时间。 示例值：2024-11-12T05:30:21Z
CreatedTime	Timestamp ISO8601	伸缩活动创建时间，为 UTC 标准时间。 示例值：2024-11-12T05:30:21Z
StatusMessageSimplified	String	伸缩活动状态简要描述。

名称	类型	描述
		示例值: Success
LifecycleActionResultSet	Array of LifecycleActionResultInfo	伸缩活动中生命周期挂钩的执行结果。 示例值: [{"NotificationResult": "NONE", "ResultReason": "HEARTBEAT_TIMEOUT", "InstanceId": "ins-1dcgn84h", "InvocationId": "", "LifecycleActionResult": "CONTINUE", "InvokeCommandResult": "NONE", "LifecycleHookId": "ash-aadtetfm"}]
DetailedStatusMessageSet	Array of DetailedStatusMessage	伸缩活动状态详细描述。
InvocationResultSet	Array of InvocationResult	执行命令结果。 示例值: [{"InstanceId": "ins-ju6vr4d3", "InvocationId": "inv-1sgbq5c1", "InvocationTaskId": "invt-6x4li1zf", "CommandId": "cmd-rkk9m8rp", "TaskStatus": "SUCCESS", "ErrorMessage": ""}]
RelatedInstanceSet	Array of RelatedInstance	伸缩活动相关实例信息集合。 示例值: [{"InstanceId": "ins-qdve9yyr", "InstanceStatus": "SUCCESSFUL"}]

Advice

伸缩配置建议。

被如下接口引用: DescribeAutoScalingAdvices。

名称	类型	描述
Problem	String	问题描述。 示例值: DiskSoldOut
Detail	String	问题详情。 示例值: Load balancers ['1b-p4nq0x9e'] are not found.
Solution	String	建议解决方案。 示例值: It is recommended to config Auto Scaling Group with valid load balancer.
Level	String	伸缩建议警告级别。取值范围: <ul style="list-style-type: none"> WARNING: 警告级别 CRITICAL: 严重级别 示例值: WARNING

AutoScalingAdvice

伸缩组配置建议。

被如下接口引用: DescribeAutoScalingAdvices。

名称	类型	描述
AutoScalingGroupId	String	伸缩组ID。 示例值: asg-pc2oqu2z
Level	String	伸缩组警告级别。取值范围: <ul style="list-style-type: none"> NORMAL: 正常 WARNING: 警告级别 CRITICAL: 严重级别 示例值: NORMAL

名称	类型	描述
Advices	Array of Advice	伸缩组配置建议集合。 示例值: [{"Problem": "InvalidInstanceType", "Detail": "Instance Type SN3ne.LARGE4(POSTPAID_BY_HOUR) in ap-shanghai-5 is invalid.", "Solution": "It is recommended to replace the invalid instance type.", "Level": "WARNING"}]

AutoScalingGroup

伸缩组

被如下接口引用: DescribeAutoScalingGroups。

名称	类型	描述
AutoScalingGroupId	String	伸缩组ID 示例值: asg-30wif85u
AutoScalingGroupName	String	伸缩组名称 示例值: test-asg
AutoScalingGroupStatus	String	伸缩组当前状态。取值范围: <ul style="list-style-type: none"> NORMAL: 正常 CVM_ABNORMAL: 启动配置异常 LB_ABNORMAL: 负载均衡器异常 LB_LISTENER_ABNORMAL: 负载均衡器监听器异常 LB_LOCATION_ABNORMAL: 负载均衡器监听器转发配置异常 VPC_ABNORMAL: VPC网络异常 SUBNET_ABNORMAL: VPC子网异常 INSUFFICIENT_BALANCE: 余额不足 LB_BACKEND_REGION_NOT_MATCH: CLB实例后端地域与AS服务所在地域不匹配 LB_BACKEND_VPC_NOT_MATCH: CLB实例VPC与伸缩组VPC不匹配 示例值: NORMAL
CreatedTime	Timestamp ISO8601	创建时间, 采用UTC标准计时 示例值: 2021-05-31T12:38:35Z
DefaultCooldown	Integer	默认冷却时间, 单位秒 示例值: 300
DesiredCapacity	Integer	期望实例数 示例值: 0
EnabledStatus	String	启用状态, 取值包括ENABLED和DISABLED 示例值: ENABLED
ForwardLoadBalancerSet	Array of ForwardLoadBalancer	应用型负载均衡器列表 示例值: [{"LoadBalancerId": "lb-af8li2kn", "ListenerId": "lbi-akw1lgo7", "TargetAttributes": [{"Port": 80, "Weight": 10}]}]
InstanceCount	Integer	实例数量 示例值: 0
InServiceInstanceCount	Integer	状态为IN_SERVICE实例的数量 示例值: 0
LaunchConfigurationId	String	启动配置ID 示例值: asc-966b7fk2

名称	类型	描述
LaunchConfigurationName	String	启动配置名称 示例值: test-lc
LoadBalancerIdSet	Array of String	传统型负载均衡器ID列表 示例值: ["lb-ij7ytg54"]
MaxSize	Integer	最大实例数 示例值: 10
MinSize	Integer	最小实例数 示例值: 0
ProjectId	Integer	项目ID 示例值: 0
SubnetIdSet	Array of String	子网ID列表 示例值: ["subnet-rdm629r2"]
TerminationPolicySet	Array of String	销毁策略。取值范围如下: <ul style="list-style-type: none"> • OLDEST_INSTANCE: 优先销毁伸缩组中最旧的实例, 默认取值。 • NEWEST_INSTANCE: 优先销毁伸缩组中最新的实例。 示例值: OLDEST_INSTANCE
VpcId	String	私有网络ID。 示例值: vpc-h3cd5td5
ZoneSet	Array of String	可用区列表 示例值: ["ap-guangzhou-5"]
RetryPolicy	String	重试策略, 部分成功的伸缩活动判定为一次失败活动。取值范围如下: <ul style="list-style-type: none"> • IMMEDIATE_RETRY: 默认取值, 表示立即重试, 在较短时间内快速重试, 连续失败超过一定次数(5次)后不再重试。 • INCREMENTAL_INTERVALS: 间隔递增重试, 随着连续失败次数的增加, 重试间隔逐渐增大。前10次重试为快速重试, 后续重试间隔逐步递增至10分钟、30分钟、60分钟、一天。 • NO_RETRY, 不进行重试, 直到再次收到用户调用或者告警信息后才会重试。 示例值: NO_RETRY
InActivityStatus	String	伸缩组是否处于伸缩活动中, IN_ACTIVITY表示处于伸缩活动中, NOT_IN_ACTIVITY表示不处于伸缩活动中。 示例值: IN_ACTIVITY
Tags	Array of Tag	伸缩组标签列表 示例值: [{"Key": "Service_B", "Value": "pss3", "ResourceType": "auto-scaling-group"}]
ServiceSettings	ServiceSettings	服务设置 示例值: {"ReplaceMonitorUnhealthy": false, "ReplaceLoadBalancerUnhealthy": false, "ReplaceMode": "RECREATE", "ScalingMode": "CLASSIC_SCALING"}
Ipv6AddressCount	Integer	实例具有IPv6地址数量的配置, 取值包括0、1。默认值为0, 表示实例不分配IPv6地址。需使用支持IPv6的私有网络, 需在子网中开启IPv6 CIDR, 其他使用限制可参考 IPv6使用限制 。 示例值: 0
MultiZoneSubnetPolicy	String	多可用区/子网策略。

名称	类型	描述
		<ul style="list-style-type: none"> PRIORITY, 按照可用区/子网列表的顺序, 作为优先级来尝试创建实例, 如果优先级最高的可用区/子网可以创建成功, 则总在该可用区/子网创建。 EQUALITY: 每次选择当前实例数最少的可用区/子网进行扩容, 使得每个可用区/子网都有机会发生扩容, 多次扩容出的实例会打散到多个可用区/子网。 示例值: PRIORITY
HealthCheckType	String	伸缩组实例健康检查类型, 取值如下: <ul style="list-style-type: none"> CVM: 根据实例网络状态判断实例是否处于不健康状态, 不健康的网络状态即发生实例 PING 不可达事件, 详细判断标准可参考实例健康检查 CLB: 根据 CLB 的健康检查状态判断实例是否处于不健康状态, CLB健康检查原理可参考健康检查 示例值: CLB
LoadBalancerHealthCheckGracePeriod	Integer	CLB健康检查宽限期.当扩容的实例进入IN_SERVICE后, 在宽限期时间范围内将不会被标记为不健康 CLB_UNHEALTHY。 默认值: 0。取值范围[0, 7200], 单位: 秒。 示例值: 300
InstanceAllocationPolicy	String	实例分配策略, 取值包括 LAUNCH_CONFIGURATION 和 SPOT_MIXED。 <ul style="list-style-type: none"> LAUNCH_CONFIGURATION, 代表传统的按照启动配置模式。 SPOT_MIXED, 代表竞价混合模式。目前仅支持启动配置为按量计费模式时使用混合模式, 混合模式下, 伸缩组将根据设定扩容按量或竞价机型。使用混合模式时, 关联的启动配置的计费类型不可被修改。 示例值: LAUNCH_CONFIGURATION
SpotMixedAllocationPolicy	SpotMixedAllocationPolicy	竞价混合模式下, 各计费类型实例的分配策略。仅当 InstanceAllocationPolicy 取 SPOT_MIXED 时才会返回有效值。 示例值: {"BaseCapacity": 1, "CompensateWithBaseInstance": true, "OnDemandPercentageAboveBaseCapacity": 50, "SpotAllocationStrategy": "COST_OPTIMIZED"}
CapacityRebalance	Boolean	容量重平衡功能, 仅对伸缩组内的竞价实例有效。取值范围: <ul style="list-style-type: none"> TRUE, 开启该功能, 当伸缩组内的竞价实例即将被竞价实例服务自动回收前, AS 主动发起竞价实例销毁流程, 如果有配置过缩容 hook, 则销毁前 hook 会生效。销毁流程启动后, AS 会异步开启一个扩容活动, 用于补齐期望实例数。 FALSE, 不开启该功能, 则 AS 等待竞价实例被销毁后才会去扩容补齐伸缩组期望实例数。 示例值: false
InstanceNameIndexSettings	InstanceNameIndexSettings	实例名称序号相关设置。 示例值: {"BeginIndex": 0, "Enabled": true}

AutoScalingGroupAbstract

伸缩组简明信息。

被如下接口引用: DescribeLaunchConfigurations。

名称	类型	描述
AutoScalingGroupId	String	伸缩组ID。 示例值：asg-djy6etgf
AutoScalingGroupName	String	伸缩组名称。 示例值：as-group-a0

AutoScalingNotification

弹性伸缩事件通知

被如下接口引用：DescribeNotificationConfigurations。

名称	类型	描述
AutoScalingGroupId	String	伸缩组ID。 示例值：asg-19kw47wk
NotificationUserGroupIds	Array of String	用户组ID列表。 示例值：["12529"]
NotificationTypes	Array of String	通知事件列表。取值范围如下： <ul style="list-style-type: none"> SCALE_OUT_SUCCESSFUL：扩容成功 SCALE_OUT_FAILED：扩容失败 SCALE_IN_SUCCESSFUL：缩容成功 SCALE_IN_FAILED：缩容失败 REPLACE_UNHEALTHY_INSTANCE_SUCCESSFUL：替换不健康子机成功 REPLACE_UNHEALTHY_INSTANCE_FAILED：替换不健康子机失败 示例值：["SCALE_OUT_SUCCESSFUL"]
AutoScalingNotificationId	String	事件通知ID。 示例值：asn-6r4waxc4
TargetType	String	通知接收端类型。取值范围如下： <ul style="list-style-type: none"> USER_GROUP：用户组 TDMQ_CMQ_TOPIC：TDMQ CMQ 主题 TDMQ_CMQ_QUEUE：TDMQ CMQ 队列 CMQ_QUEUE：CMQ 队列，CMQ 接口已下线，已无法选用 CMQ_TOPIC：CMQ 主题，CMQ 接口已下线，已无法选用 示例值：USER_GROUP
QueueName	String	TDMQ CMQ 队列名。 示例值：test-queue
TopicName	String	TDMQ CMQ 主题名。 示例值：test-topic

DataDisk

启动配置的数据盘配置信息。若不指定该参数，则默认不购买数据盘，当前仅支持购买的时候指定一个数据盘。

被如下接口引用：CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
DiskType	String	否	数据盘类型。数据盘类型限制详见 云硬盘类型 。取值范围： <ul style="list-style-type: none"> LOCAL_BASIC：本地硬盘 LOCAL_SSD：本地SSD硬盘 CLOUD_BASIC：普通云硬盘 CLOUD_PREMIUM：高性能云硬盘 CLOUD_SSD：SSD云硬盘

名称	类型	必选	描述
			<ul style="list-style-type: none"> • CLOUD_HSSD: 增强型SSD云硬盘 • CLOUD_TSSD: 极速型SSD云硬盘 • CLOUD_BSSD: 通用型SSD云硬盘 默认取值与系统盘类型 (SystemDisk.DiskType) 保持一致。 示例值: CLOUD_PREMIUM
DiskSize	Integer	否	数据盘大小, 单位: GB。不同数据盘类型取值范围不同, 具体限制详见: CVM实例配置 。默认值为0, 表示不购买数据盘。更多限制详见 产品文档 。 示例值: 0
SnapshotId	String	否	数据盘快照 ID, 可通过 DescribeSnapshots 接口获取该参数。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: snap-l8psqwnt
DeleteWithInstance	Boolean	否	数据盘是否随子机销毁。取值范围: <ul style="list-style-type: none"> • TRUE: 子机销毁时, 销毁数据盘, 只支持按小时后付费云盘 • FALSE: 子机销毁时, 保留数据盘 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: false
Encrypt	Boolean	否	数据盘是否加密。取值范围: <ul style="list-style-type: none"> • TRUE: 加密 • FALSE: 不加密 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: true
ThroughputPerformance	Integer	否	云硬盘性能, 单位: MB/s。使用此参数可给云硬盘购买额外的性能, 功能介绍和类型限制详见: 增强型 SSD 云硬盘额外性能说明 。 当前仅支持极速型云盘 (CLOUD_TSSD) 和增强型SSD云硬盘 (CLOUD_HSSD) 且需容量 > 460GB。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 100
BurstPerformance	Boolean	否	突发性能。是否开启突发性能, 默认取值为 false。当前该参数仅支持极速型云盘 (CLOUD_TSSD) 和增强型SSD云硬盘 (CLOUD_HSSD) 且需容量 > 460GB。 注: 内测中, 需提单申请后使用。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: false

DetailedStatusMessage

伸缩活动状态详细描述。

被如下接口引用: DescribeAutoScalingActivities, DescribeAutoScalingGroupLastActivities。

名称	类型	描述
Code	String	错误类型。 示例值: InvalidInstanceType
Zone	String	可用区信息。 示例值: ap-guangzhou-2
InstanceId	String	实例ID。 示例值: ins-yy23daou
InstanceChargeType	String	实例计费类型。 示例值: POSTPAID_BY_HOUR
SubnetId	String	子网ID。

名称	类型	描述
		示例值: subnet-gz7j7r3k
Message	String	错误描述。 示例值: Failed to launch instance ins-9ojy714a. Instance type: S2.MEDIUM2, Subnet: subnet-6400tdiy, Zone: ap-guangzhou-2, LatestOperationCode: ResourceInsufficient.
InstanceType	String	实例类型。 示例值: IT5.8XLARGE128

EnhancedService

描述了实例的增强服务启用情况与其设置，如云安全，云监控，自动化助手等实例 Agent。

被如下接口引用：CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
SecurityService	RunSecurityServiceEnabled	否	开启云安全服务。若不指定该参数，则默认开启云安全服务。 示例值: {"Enabled": True}
MonitorService	RunMonitorServiceEnabled	否	开启云监控服务。若不指定该参数，则默认开启云监控服务。 示例值: {"Enabled": True}
AutomationToolsService	RunAutomationServiceEnabled	否	开启自动化助手服务。若不指定该参数，则默认逻辑与CVM保持一致。注意：此字段可能返回 null，表示取不到有效值。 示例值: {"Enabled": True}

Filter

描述键值对过滤器，用于条件过滤查询。例如过滤ID、名称、状态等

- 若存在多个Filter时，Filter间的关系为逻辑与（AND）关系。
- 若同一个Filter存在多个Values，同一Filter下Values间的关系为逻辑或（OR）关系。

以DescribeInstances接口的Filter为例。若我们需要查询可用区（zone）为广州一区 并且实例计费模式（instance-charge-type）为包年包月 或者 按量计费的实例时，可如下实现：

```
Filters.0.Name=zone
&Filters.0.Values.0=ap-guangzhou-1
&Filters.1.Name=instance-charge-type
&Filters.1.Values.0=PREPAID
&Filters.1.Values.1=POSTPAID_BY_HOUR
```

被如下接口引用：DescribeAutoScalingActivities, DescribeAutoScalingGroups, DescribeAutoScalingInstances, DescribeLaunchConfigurations, DescribeLifecycleHooks, DescribeNotificationConfigurations, DescribeRefreshActivities, DescribeScalingPolicies, DescribeScheduledActions。

名称	类型	必选	描述
Name	String	是	需要过滤的字段。 示例值: zone
Values	Array of String	是	字段的过滤值。 示例值: ap-guangzhou-6

ForwardLoadBalancer

应用型负载均衡器

被如下接口引用：AttachLoadBalancers, CreateAutoScalingGroup, DescribeAutoScalingGroups, ModifyLoadBalancerTargetAttributes, ModifyLoadBalancers。

名称	类型	必选	描述
LoadBalancerId	String	是	负载均衡器ID。作为入参时，该参数必填。可以通过 DescribeLoadBalancers 接口获取。 示例值：lb-fy6te4r5
ListenerId	String	是	负载均衡监听器 ID。作为入参时，该参数必填。可以通过 DescribeLoadBalancers 接口获取。 示例值：lbl-dhyt56rf
TargetAttributes	Array of TargetAttribute	是	目标规则属性列表。作为入参时，该参数必填。 示例值：[{"Port": 80, "Weight": 10}]
LocationId	String	否	转发规则ID，注意：针对七层监听器此参数必填。可以通过 DescribeLoadBalancers 接口获取。 示例值：loc-sj8y6tr4
Region	String	否	负载均衡实例所属地域，默认取AS服务所在地域。格式与公共参数Region相同，如："ap-guangzhou"。 示例值：ap-guangzhou

ForwardLoadBalancerIdentification

应用型负载均衡器标识信息

被如下接口引用：DetachLoadBalancers。

名称	类型	必选	描述
LoadBalancerId	String	是	负载均衡器ID 示例值：lb-ceqnq5xk
ListenerId	String	是	应用型负载均衡监听器 ID 示例值：lbl-lqf523ih
LocationId	String	否	转发规则ID，注意：针对七层监听器此参数必填 示例值：loc-0xbavbfi

HostNameSettings

云服务器主机名（HostName）的相关设置

被如下接口引用：CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
HostName	String	是	<p>云服务器的主机名。</p> <ul style="list-style-type: none"> 点号（.）和短横线（-）不能作为 HostName 的首尾字符，不能连续使用。 不支持 Windows 实例。 其他类型（Linux 等）实例：字符长度为[2, 40]，允许支持多个点号，点之间为一段，每段允许字母（不限制大小写）、数字和短横线（-）组成。不允许为纯数字。 <p>注意：此字段可能返回 null，表示取不到有效值。 示例值：product-as-host-0</p>

名称	类型	必选	描述
HostNameStyle	String	否	<p>云服务器主机名的风格，取值范围包括 ORIGINAL 和 UNIQUE，默认为 ORIGINAL。</p> <ul style="list-style-type: none"> ORIGINAL，AS 直接将入参中所填的 HostName 传递给 CVM，CVM 可能会对 HostName 追加序号，伸缩组中实例的 HostName 会出现冲突的情况。 UNIQUE，入参所填的 HostName 相当于主机名前缀，AS 和 CVM 会对其进行拓展，伸缩组中实例的 HostName 可以保证唯一。 <p>注意：此字段可能返回 null，表示取不到有效值。 示例值：ORIGINAL</p>
HostNameSuffix	String	否	<p>云服务器的主机名后缀。</p> <ul style="list-style-type: none"> 点号 (.) 和短横线 (-) 不能作为 HostNameSuffix 的首尾字符，不能连续使用。 不支持 Windows 实例。 其他类型 (Linux 等) 实例：字符长度为 [1, 37]，且与 HostName 的长度和不能超过 39，允许支持多个点号，点之间为一段，每段允许字母 (不限制大小写)、数字和短横线 (-) 组成。 <p>假设后缀名称为 suffix，原主机名为 test.0，最终主机名为 test.0.suffix。 注意：此字段可能返回 null，表示取不到有效值。 示例值：suffix</p>

IPv6InternetAccessible

描述了启动配置创建实例的IPv6地址公网可访问性，声明了IPv6地址公网使用计费模式，最大带宽等

被如下接口引用：CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
InternetChargeType	String	否	<p>网络计费模式。取值包括TRAFFIC_POSTPAID_BY_HOUR、BANDWIDTH_PACKAGE，默认取值为TRAFFIC_POSTPAID_BY_HOUR。查看当前账户类型可参考账户类型说明。</p> <ul style="list-style-type: none"> IPv6对标准账户类型支持TRAFFIC_POSTPAID_BY_HOUR。 IPv6对传统账户类型支持BANDWIDTH_PACKAGE。 <p>注意：此字段可能返回 null，表示取不到有效值。 示例值：TRAFFIC_POSTPAID_BY_HOUR</p>
InternetMaxBandwidthOut	Integer	否	<p>公网出带宽上限，单位：Mbps。 默认值：0，此时不为IPv6分配公网带宽。不同机型、可用区、计费模式的带宽上限范围不一致，具体限制详见公网带宽上限。</p> <p>注意：此字段可能返回 null，表示取不到有效值。 示例值：0</p>
BandwidthPackageId	String	否	<p>带宽包ID。可通过DescribeBandwidthPackages接口返回值中的BandwidthPackageId获取。</p> <p>注意：此字段可能返回 null，表示取不到有效值。 示例值：bwp-jfio9i0p</p>

Instance

实例信息

被如下接口引用：DescribeAutoScalingInstances。

名称	类型	描述
InstanceId	String	实例ID 示例值: ins-089dlszs
AutoScalingGroupId	String	伸缩组ID 示例值: asg-02ofd40a
LaunchConfigurationId	String	启动配置ID 示例值: asc-23yf98ie
LaunchConfigurationName	String	启动配置名称 示例值: 测试asc-01_2.3
LifeCycleState	String	生命周期状态, 取值如下: <ul style="list-style-type: none"> IN_SERVICE: 运行中 CREATING: 创建中 CREATION_FAILED: 创建失败 TERMINATING: 中止中 TERMINATION_FAILED: 中止失败 ATTACHING: 绑定中 ATTACH_FAILED: 绑定失败 DETACHING: 解绑中 DETACH_FAILED: 解绑失败 ATTACHING_LB: 绑定LB中 DETACHING_LB: 解绑LB中 MODIFYING_LB: 修改LB中 STARTING: 开机中 START_FAILED: 开机失败 STOPPING: 关机中 STOP_FAILED: 关机失败 STOPPED: 已关机 IN_LAUNCHING_HOOK: 扩容生命周期挂钩中 IN_TERMINATING_HOOK: 缩容生命周期挂钩中 示例值: IN_SERVICE
HealthStatus	String	健康状态, 取值范围如下: <ul style="list-style-type: none"> HEALTHY: 实例处于健康状态 UNHEALTHY: 实例 ping 不可达 CLB_UNHEALTHY: CLB 监听的实例端口不健康 示例值: HEALTHY
ProtectedFromScaleIn	Boolean	是否加入缩容保护 示例值: false
Zone	String	可用区 示例值: ap-guangzhou-1
CreationType	String	创建类型, 取值包括AUTO_CREATION, MANUAL_ATTACHING。 示例值: AUTO_CREATION
AddTime	Timestamp ISO8601	实例加入时间, 按照ISO8601标准表示, 并且使用UTC时间。 示例值: 2023-01-13T03:06:25Z
InstanceType	String	实例类型 示例值: S3.MEDIUM2
VersionNumber	Integer	版本号 示例值: 1
AutoScalingGroupName	String	伸缩组名称 示例值: 测试asg-01_2.3
WarmupStatus	String	预热状态, 取值如下:

名称	类型	描述
		<ul style="list-style-type: none"> WAITING_ENTER_WARMUP: 等待进入预热 NO_NEED_WARMUP: 无需预热 IN_WARMUP: 预热中 AFTER_WARMUP: 完成预热 示例值: IN_WARMUP
DisasterRecoverGroupIds	Array of String	置放群组id, 仅支持指定一个。 示例值: ["ps-uf8j6y3t"]

InstanceChargePrepaid

描述了实例的计费模式

被如下接口引用: CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
Period	Integer	是	购买实例的时长, 单位: 月。取值范围: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36。 示例值: 1
RenewFlag	String	否	自动续费标识。取值范围: <ul style="list-style-type: none"> NOTIFY_AND_AUTO_RENEW: 通知过期且自动续费 NOTIFY_AND_MANUAL_RENEW: 通知过期不自动续费 DISABLE_NOTIFY_AND_MANUAL_RENEW: 不通知过期不自动续费 默认取值: NOTIFY_AND_MANUAL_RENEW。若该参数指定为NOTIFY_AND_AUTO_RENEW, 在账户余额充足的情况下, 实例到期后将按月自动续费。 示例值: NOTIFY_AND_MANUAL_RENEW

InstanceMarketOptionsRequest

CVM竞价请求相关选项

被如下接口引用: CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
SpotOptions	SpotMarketOptions	是	竞价相关选项 示例值: {"SpotInstanceType": "one-time", "MaxPrice": 1000}
MarketType	String	否	市场选项类型, 当前只支持取值: spot 示例值: spot

InstanceNameIndexSettings

实例名称序号相关设置。

被如下接口引用: CreateAutoScalingGroup, DescribeAutoScalingGroups, ModifyAutoScalingGroup。

名称	类型	必选	描述
Enabled	Boolean	否	是否开启实例创建序号, 默认不开启。取值范围: <ul style="list-style-type: none"> TRUE: 表示开启实例创建序号 FALSE: 表示不开启实例创建序号 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: FALSE
BeginIndex	Integer	否	初始序号, 取值范围为 [0, 99999999]。当序号递增后超出取值范围时, 扩容活动会失败。 <ul style="list-style-type: none"> 首次开启实例名称序号: 默认为 0。

名称	类型	必选	描述
			<ul style="list-style-type: none"> 非首次开启实例名称序号：若不指定该参数，沿用历史序号。下调初始序号可能会造成伸缩组内实例名称序号重复。 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

InstanceNameSettings

云服务器实例名称（InstanceName）的相关设置

被如下接口引用：CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
InstanceName	String	是	云服务器的实例名。字符长度为[2, 108]。 示例值：product-as-ins-0
InstanceNameStyle	String	否	云服务器实例名的风格，取值范围包括 ORIGINAL 和 UNIQUE，默认为 ORIGINAL。 ORIGINAL，AS 直接将入参中所填的 InstanceName 传递给 CVM，CVM 可能会对 InstanceName 追加序列号，伸缩组中实例的 InstanceName 会出现冲突的情况。 UNIQUE，入参所填的 InstanceName 相当于实例名前缀，AS 和 CVM 会对其进行拓展，伸缩组中实例的 InstanceName 可以保证唯一。 示例值：ORIGINAL
InstanceNameSuffix	String	否	云服务器实例名后缀。字符长度为[1,105]，且与 InstanceName 的长度和不能超过107。 假设后缀名称为 suffix，原实例名为 test.0，最终实例名为 test.0.suffix。 注意：此字段可能返回 null，表示取不到有效值。 示例值：testsuffix

InstanceTag

实例标签。通过指定该参数，可以为扩容的实例绑定标签。

被如下接口引用：CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
Key	String	是	标签键 示例值：as-tag-key-app0
Value	String	是	标签值 示例值：as-tag-value-app0

InternetAccessible

描述了启动配置创建实例的公网可访问性，声明了实例的公网使用计费模式，最大带宽等

被如下接口引用：CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
InternetChargeType	String	否	网络计费类型。取值范围： <ul style="list-style-type: none"> BANDWIDTH_PREPAID：预付费按带宽结算 TRAFFIC_POSTPAID_BY_HOUR：流量按小时后付费 BANDWIDTH_POSTPAID_BY_HOUR：带宽按小时后付费 BANDWIDTH_PACKAGE：带宽包用户

名称	类型	必选	描述
			默认取值：TRAFFIC_POSTPAID_BY_HOUR。 注意：此字段可能返回 null，表示取不到有效值。 示例值：TRAFFIC_POSTPAID_BY_HOUR
InternetMaxBandwidthOut	Integer	否	公网出带宽上限，单位：Mbps。默认值：0Mbps。不同机型带宽上限范围不一致，具体限制详见 购买网络带宽 。 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
PublicIpAssigned	Boolean	否	是否分配公网IP。取值范围： • TRUE：表示分配公网IP • FALSE：表示不分配公网IP 当公网带宽大于0Mbps时，可自由选择开通与否，默认开通公网IP；当公网带宽为0，则不允许分配公网IP。 注意：此字段可能返回 null，表示取不到有效值。 示例值：false
BandwidthPackageId	String	否	带宽包ID。可通过 DescribeBandwidthPackages 接口返回值中的BandwidthPackageId获取。 注意：此字段可能返回 null，表示取不到有效值。 示例值：bwp-dss86yt5

InvocationResult

执行命令结果。

被如下接口引用：DescribeAutoScalingActivities, DescribeAutoScalingGroupLastActivities。

名称	类型	描述
InstanceId	String	实例ID。 示例值：ins-0lizunus
InvocationId	String	执行活动ID。 示例值：inv-0h9q6ahv
InvocationTaskId	String	执行任务ID。 示例值：invt-4mqj9ekr
CommandId	String	命令ID。 示例值：cmd-5l44tu7j
TaskStatus	String	执行任务状态。 示例值：SUCCESS
ErrorMessage	String	执行异常信息。 示例值：InvocationTaskFailed->Invocation task failed.

LaunchConfiguration

符合条件的启动配置信息的集合。

被如下接口引用：DescribeLaunchConfigurations。

名称	类型	描述
ProjectId	Integer	实例所属项目ID。 示例值：0
LaunchConfigurationId	String	启动配置ID。 示例值：asc-abcd0123

名称	类型	描述
LaunchConfigurationName	String	启动配置名称。 示例值：TestLauchConfig0_0
InstanceType	String	实例机型。 示例值：SA2.MEDIUM2
SystemDisk	SystemDisk	实例系统盘配置信息。 示例值：{"DiskSize": 500, "DiskType": "CLOUD_BSSD"}
DataDisks	Array of DataDisk	实例数据盘配置信息。 示例值：[{"DiskSize": 200, "DiskType": "CLOUD_BSSD"}]
LoginSettings	LimitedLoginSettings	实例登录设置。 示例值：{"KeyIds": ["skey-sm1dxnap"]}
InternetAccessible	InternetAccessible	公网带宽相关信息设置。 示例值：{"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR", "InternetMaxBandwidthOut": 0, "PublicIpAssigned": false}
SecurityGroupIds	Array of String	实例所属安全组。 示例值：["sg-brwzhm6d"]
AutoScalingGroupAbstractSet	Array of AutoScalingGroupAbstract	启动配置关联的伸缩组。 示例值：[{"AutoScalingGroupId": 'asg-1dbjno4d', 'AutoScalingGroupName': 'asg-v000'}]
UserData	String	自定义数据。 注意：此字段可能返回 null，表示取不到有效值。 示例值：dGVzdA==
CreatedTime	Timestamp ISO8601	启动配置创建时间，为标准UTC时间。 示例值：2021-05-31T12:38:35Z
EnhancedService	EnhancedService	实例的增强服务启用情况与其设置。 示例值：{"MonitorService": {"Enabled": true}, "SecurityService": {"Enabled": true}}
ImageId	String	镜像ID。 示例值：img-abcd0123
LaunchConfigurationStatus	String	启动配置当前状态。取值范围： <ul style="list-style-type: none"> NORMAL：正常 IMAGE_ABNORMAL：启动配置镜像异常 CBS_SNAP_ABNORMAL：启动配置数据盘快照异常 SECURITY_GROUP_ABNORMAL：启动配置安全组异常 示例值：NORMAL
InstanceChargeType	String	实例计费类型，取值范围如下： <ul style="list-style-type: none"> POSTPAID_BY_HOUR：按小时后付费 SPOTPAID：竞价付费 PREPAID：预付费，即包年包月 CDCPAID：专用集群付费 示例值：POSTPAID_BY_HOUR
InstanceMarketOptions	InstanceMarketOptionsRequest	实例的市场相关选项，如竞价实例相关参数，若指定实例的付费模式为竞价付费则该参数必传。

名称	类型	描述
		注意：此字段可能返回 null，表示取不到有效值。 示例值：{"SpotOptions": {"SpotInstanceType": "one-time", "MaxPrice": 1000}, "MarketType": "spot"}
InstanceTypes	Array of String	实例机型列表。 示例值：["SA2.MEDIUM2"]
InstanceTags	Array of InstanceTag	实例标签列表。扩容出来的实例会自动带上标签，最多支持10个标签。 示例值：[{"Key": "server-group-name", "Value": "as-tencent"}]
Tags	Array of Tag	标签列表，该参数内的标签仅用于绑定启动配置，不会传递给基于该启动配置扩容的 CVM 实例。 示例值：[{"Key": "CCode", "Value": "ASTENCENT", "ResourceType": "launch-configuration"}]
VersionNumber	Integer	版本号。 示例值：1
UpdateTime	Timestamp ISO8601	更新时间，为标准UTC时间。 示例值：2023-09-04T11:42:46Z
CamRoleName	String	CAM角色名称。可通过 DescribeRoleList 接口返回值中的roleName获取。 示例值：TestRole
LastOperationInstanceTypesCheckPolicy	String	上次操作时，InstanceTypesCheckPolicy 取值。 示例值：ANY
HostNameSettings	HostNameSettings	云服务器主机名（HostName）的相关设置。 示例值：{"HostNameStyle": "UNIQUE", "HostName": "as-host"}
InstanceNameSettings	InstanceNameSettings	云服务器实例名（InstanceName）的相关设置。 示例值：{"InstanceName": "app-as", "InstanceNameStyle": "UNIQUE"}
InstanceChargePrepaid	InstanceChargePrepaid	预付费模式，即包年包月相关参数设置。通过该参数可以指定包年包月实例的购买时长、是否设置自动续费等属性。若指定实例的付费模式为预付费则该参数必传。 示例值：{"Period": 1, "RenewFlag": "NOTIFY_AND_AUTO_RENEW"}
DiskTypePolicy	String	云盘类型选择策略。取值范围： • ORIGINAL：使用设置的云盘类型 • AUTOMATIC：自动选择当前可用区下可用的云盘类型 示例值：ORIGINAL
HpcClusterId	String	高性能计算集群ID。 注意：此字段默认为空。 示例值：hpc-abcd0123
IPv6InternetAccessible	IPv6InternetAccessible	IPv6公网带宽相关信息设置。 示例值：{"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR", "InternetMaxBandwidthOut": 1}
DisasterRecoverGroupIds	Array of String	置放群组id，仅支持指定一个。

名称	类型	描述
		示例值: ["ps-dy7ui89o"]
ImageFamily	String	镜像族名称。 示例值: testfamily
DedicatedClusterId	String	本地专用集群 ID。 示例值: cluster-gbo27yc4

LifecycleActionResultInfo

生命周期挂钩动作的执行结果信息。

被如下接口引用: DescribeAutoScalingActivities, DescribeAutoScalingGroupLastActivities。

名称	类型	描述
LifecycleHookId	String	生命周期挂钩标识。 示例值: ash-xxxxxxxx
InstanceId	String	实例标识。 示例值: ins-xxxxxxxx
InvocationId	String	执行活动ID。可通过TAT的 查询执行活动API 查询具体的执行结果。 示例值: inv-xxxxxxxx
InvokeCommandResult	String	命令调用的结果, 表示执行TAT命令是否成功。 <ul style="list-style-type: none"> SUCCESSFUL 命令调用成功, 不代表命令执行成功, 执行的具体情况可根据InvocationId进行查询 FAILED 命令调用失败 NONE 示例值: SUCCESSFUL
NotificationResult	String	通知的结果, 表示通知CMQ/TDMQ是否成功。 <ul style="list-style-type: none"> SUCCESSFUL 通知成功 FAILED 通知失败 NONE 示例值: null
LifecycleActionResult	String	生命周期挂钩动作的执行结果, 取值包括 CONTINUE、ABANDON。 示例值: CONTINUE
ResultReason	String	结果的原因。 <ul style="list-style-type: none"> HEARTBEAT_TIMEOUT 由于心跳超时, 结果根据DefaultResult设置。 NOTIFICATION_FAILURE 由于发送通知失败, 结果根据DefaultResult设置。 CALL_INTERFACE 调用了接口CompleteLifecycleAction设置结果。 ANOTHER_ACTION_ABANDON 另一个生命周期操作的结果已设置为“ABANDON”。 COMMAND_CALL_FAILURE 由于命令调用失败, 结果根据DefaultResult设置。 COMMAND_EXEC_FINISH 命令执行完成。 COMMAND_EXEC_FAILURE 由于命令执行失败, 结果根据DefaultResult设置。 COMMAND_EXEC_RESULT_CHECK_FAILURE 由于命令结果检查失败, 结果根据DefaultResult设置。 示例值: HEARTBEAT_TIMEOUT

LifecycleCommand

远程命令执行对象。

被如下接口引用: CreateLifecycleHook, DescribeLifecycleHooks, ModifyLifecycleHook, UpgradeLifecycleHook。

名称	类型	必选	描述
CommandId	String	是	远程命令ID。若选择执行命令, 则此项必填。 示例值: cmd-gt5dre32
Parameters	String	否	自定义参数。字段类型为 json encoded string。如: {"varA": "222"}。 key为自定义参数名称, value为该参数的默认取值。kv均为字符串型。 如果未提供该参数取值, 将使用 Command 的 DefaultParameters 进行替换。 自定义参数最多20个。自定义参数名称需符合以下规范: 字符数目上限64, 可选范围 [a-zA-Z0-9-_]。 示例值: {"varA": "222"}

LifecycleHook

生命周期挂钩

被如下接口引用: DescribeLifecycleHooks。

名称	类型	描述
LifecycleHookId	String	生命周期挂钩ID 示例值: ash-ck4bkcyj
LifecycleHookName	String	生命周期挂钩名称 示例值: as-hook0
AutoScalingGroupId	String	伸缩组ID 示例值: asg-lk5nn4xi
DefaultResult	String	定义伸缩组在生命周期挂钩超时或 LifecycleCommand 执行失败时应采取的操作, 取值范围如下: - CONTINUE: 默认值, 表示继续执行扩容活动 - ABANDON: 针对扩容挂钩, 挂钩超时或 LifecycleCommand 执行失败的 CVM 实例会直接释放或移出; 而针对缩容挂钩, 会继续执行缩容活动 示例值: CONTINUE
HeartbeatTimeout	Integer	生命周期挂钩超时等待时间 (以秒为单位), 范围从 30 到 7200 秒。 示例值: 300
LifecycleTransition	String	生命周期挂钩场景, 取值范围如下: - INSTANCE_LAUNCHING: 扩容生命周期挂钩 - INSTANCE_TERMINATING: 缩容生命周期挂钩 示例值: INSTANCE_TERMINATING
NotificationMetadata	String	通知目标的附加信息 示例值: as-metadata-0
CreatedTime	Timestamp ISO8601	创建时间, 采用 UTC 标准计时 示例值: 2024-11-11T15:29:59Z
NotificationTarget	NotificationTarget	通知目标 示例值: {"QueueName": "as-pool", "TargetType": "TDMQ_CMQ_QUEUE"}
LifecycleTransitionType	String	进行生命周期挂钩的场景类型, 取值范围包括 NORMAL 和 EXTENSION, 默认值为 NORMAL。 说明: 设置为EXTENSION值, 在AttachInstances、DetachInstances、RemoveInstances 接口时会触发生命周期挂钩操作, 值为NORMAL则不会在这些接口中触发生命周期挂钩。 示例值: NORMAL
LifecycleCommand	LifecycleCommand	远程命令执行对象 示例值: {"CommandId": "cmd-am4e3hd8", "Parameters": "{123: 222}"}

LimitedLoginSettings

描述了实例登录相关配置与信息，出于安全性考虑，不会描述敏感信息。

被如下接口引用：DescribeLaunchConfigurations。

名称	类型	描述
KeyIds	Array of String	密钥ID列表。 示例值：["skey-jud75er4"]

LoginSettings

描述了实例登录相关配置与信息。

被如下接口引用：CreateLaunchConfiguration, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
Password	String	否	实例登录密码。不同操作系统类型密码复杂度限制不一样，具体如下： - Linux实例密码必须8到30位，至少包括四项（小写字母、大写字母、数字和特殊符号）中的两项。 - Windows实例密码必须12到30位，至少包括包括四项（小写字母、大写字母、数字和特殊符号）中的三项。 - 若不指定该参数，则由系统随机生成密码，并通过站内信方式通知到用户。 特殊符号的取值范围：[() ` ~ ! @ # \$ % ^ & * - + = { } [] ; ' , . ? /] 示例值：passExam@!7862543
KeyIds	Array of String	否	密钥ID列表。关联密钥后，就可以通过对应的私钥来访问实例；KeyId可通过接口 DescribeKeyPairs 获取，密钥与密码不能同时指定，同时Windows操作系统不支持指定密钥。当前仅支持购买的时候指定一个密钥。 示例值：["skey-ju7fy6t5"]
KeepImageLogin	Boolean	否	保持镜像的原始设置。该参数与Password或KeyIds.N不能同时指定。只有使用自定义镜像、共享镜像或外部导入镜像创建实例时才能指定该参数为TRUE。取值范围： • TRUE：表示保持镜像的登录设置 • FALSE：表示不保持镜像的登录设置 默认取值：FALSE。 示例值：false

Metadata

自定义 Metadata

被如下接口引用：CreateLaunchConfiguration, ModifyLaunchConfigurationAttributes。

名称	类型	必选	描述
Items	Array of MetadataItem	否	自定义 Metadata 键值对列表 示例值：[{"Key": "Key123_4-5", "Value": "valueExam"}]

MetadataItem

自定义 Metadata 的一组键值对信息

被如下接口引用：CreateLaunchConfiguration, ModifyLaunchConfigurationAttributes。

名称	类型	必选	描述
Key	String	是	自定义 MetaData 键 示例值：keyA-123_4
Value	String	是	自定义 MetaData 值 示例值：Value123

MetricAlarm

弹性伸缩告警指标

被如下接口引用: CreateScalingPolicy, DescribeScalingPolicies, ModifyScalingPolicy。

名称	类型	必选	描述
ComparisonOperator	String	是	比较运算符, 可选值: <ul style="list-style-type: none"> GREATER_THAN: 大于 GREATER_THAN_OR_EQUAL_TO: 大于或等于 LESS_THAN: 小于 LESS_THAN_OR_EQUAL_TO: 小于或等于 EQUAL_TO: 等于 NOT_EQUAL_TO: 不等于 示例值: GREATER_THAN
MetricName	String	是	指标名称, 可选字段如下: <ul style="list-style-type: none"> CPU_UTILIZATION: CPU利用率 MEM_UTILIZATION: 内存利用率 LAN_TRAFFIC_OUT: 内网出带宽 LAN_TRAFFIC_IN: 内网入带宽 WAN_TRAFFIC_OUT: 外网出带宽 WAN_TRAFFIC_IN: 外网入带宽 TCP_CURR_ESTAB: TCP连接数 示例值: CPU_UTILIZATION
Threshold	Integer	是	告警阈值: <ul style="list-style-type: none"> CPU_UTILIZATION: [1, 100], 单位: % MEM_UTILIZATION: [1, 100], 单位: % LAN_TRAFFIC_OUT: >0, 单位: Mbps LAN_TRAFFIC_IN: >0, 单位: Mbps WAN_TRAFFIC_OUT: >0, 单位: Mbps WAN_TRAFFIC_IN: >0, 单位: Mbps TCP_CURR_ESTAB: >0, 单位: Count 示例值: 10
Period	Integer	是	时间周期, 单位: 秒, 取值枚举值为60、300。 示例值: 60
ContinuousTime	Integer	是	重复次数。取值范围 [1, 10] 示例值: 3
Statistic	String	否	统计类型, 可选字段如下: <ul style="list-style-type: none"> AVERAGE: 平均值 MAXIMUM: 最大值 MINIMUM: 最小值 默认取值: AVERAGE 示例值: AVERAGE
PreciseThreshold	Float	否	精确告警阈值, 本参数不作为入参输入, 仅用作查询接口出参: <ul style="list-style-type: none"> CPU_UTILIZATION: (0, 100], 单位: % MEM_UTILIZATION: (0, 100], 单位: % LAN_TRAFFIC_OUT: >0, 单位: Mbps LAN_TRAFFIC_IN: >0, 单位: Mbps WAN_TRAFFIC_OUT: >0, 单位: Mbps WAN_TRAFFIC_IN: >0, 单位: Mbps TCP_CURR_ESTAB: >0, 单位: Count 示例值: 10.123

NotificationTarget

通知目标

被如下接口引用：CreateLifecycleHook, DescribeLifecycleHooks, ModifyLifecycleHook, UpgradeLifecycleHook。

名称	类型	必选	描述
TargetType	String	是	目标类型，取值范围包括CMQ_QUEUE、CMQ_TOPIC、TDMQ_CMQ_QUEUE、TDMQ_CMQ_TOPIC。 • CMQ_QUEUE，指腾讯云消息队列-队列模型，对应产品已下线， 建议切换 TDMQ_CMQ_QUEUE 使用。 • CMQ_TOPIC，指腾讯云消息队列-主题模型，对应产品已下线， 建议切换 TDMQ_CMQ_TOPIC 使用。 • TDMQ_CMQ_QUEUE，指腾讯云 TDMQ 消息队列-队列模型。 • TDMQ_CMQ_TOPIC，指腾讯云 TDMQ 消息队列-主题模型。 示例值：TDMQ_CMQ_QUEUE
QueueName	String	否	队列名称，如果TargetType取值为CMQ_QUEUE 或 TDMQ_CMQ_QUEUE，则本字段必填。 示例值：test-queue
TopicName	String	否	主题名称，如果TargetType取值为CMQ_TOPIC 或 TDMQ_CMQ_TOPIC，则本字段必填。 示例值：test-topic

RefreshActivity

实例刷新活动。

被如下接口引用：DescribeRefreshActivities。

名称	类型	描述
AutoScalingGroupId	String	伸缩组 ID。 示例值：asg-t67yhy7u
RefreshActivityId	String	刷新活动 ID。 示例值：asr-v8dd7er6
OriginRefreshActivityId	String	原始刷新活动ID，仅在回滚刷新活动中存在。 示例值：asr-v8dd7er6
RefreshBatchSet	Array of RefreshBatch	刷新批次信息列表。 示例值：[{"RefreshBatchNum": 1, "RefreshBatchRelatedInstanceSet": [{"InstanceId": "ins-m30w46cu", "InstanceStatus": "SUCCESSFUL"}], "StartTime": "2024-11-06T19:00:14Z", "RefreshBatchStatus": "SUCCESSFUL"}]
RefreshMode	String	刷新模式。取值范围如下： • ROLLING_UPDATE_RESET：重装系统进行滚动更新 • ROLLING_UPDATE_REPLACE：新建实例替换进行滚动更新，该模式暂不支持回滚接口 示例值：ROLLING_UPDATE_RESET
RefreshSettings	RefreshSettings	实例更新设置参数。 示例值：{"CheckInstanceTargetHealth": true, "RollingUpdateSettings": {"BatchNumber": 1, "MaxSurge": 1, "BatchPause": "AUTOMATIC"}}
ActivityType	String	刷新活动类型。取值如下： • NORMAL：正常刷新活动 • ROLLBACK：回滚刷新活动 示例值：NORMAL
Status	String	刷新活动状态。取值如下： • INIT：初始化中 • RUNNING：运行中

名称	类型	描述
		<ul style="list-style-type: none"> • SUCCESSFUL: 活动成功 • FAILED_PAUSE: 因刷新批次失败暂停 • AUTO_PAUSE: 因暂停策略自动暂停 • MANUAL_PAUSE: 手动暂停 • CANCELLED: 活动取消 • FAILED: 活动失败 示例值: RUNNING
CurrentRefreshBatchNum	Integer	当前刷新批次序号。例如, 2 表示当前活动正在刷新第二批次的实例。 示例值: 2
StartTime	Timestamp ISO8601	刷新活动开始时间, 为标准 UTC 时间, 格式形如 YYYY-MM-DDTHH:mm:ssZ。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-09-04T11:42:46Z
EndTime	Timestamp ISO8601	刷新活动结束时间, 为标准 UTC 时间, 格式形如 YYYY-MM-DDTHH:mm:ssZ。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-09-04T11:42:46Z
CreatedTime	Timestamp ISO8601	刷新活动创建时间, 为标准 UTC 时间, 格式形如 YYYY-MM-DDTHH:mm:ssZ。 示例值: 2023-09-04T11:42:46Z

RefreshBatch

实例刷新批次信息, 包含该批次的刷新状态、实例、起止时间等信息。

被如下接口引用: DescribeRefreshActivities。

名称	类型	描述
RefreshBatchNum	Integer	刷新批次序号。例如, 2 表示当前批次实例会在第二批次进行实例刷新。 示例值: 2
RefreshBatchStatus	String	刷新批次状态。取值如下: <ul style="list-style-type: none"> • WAITING: 待刷新 • INIT: 初始化中 • RUNNING: 刷新中 • FAILED: 刷新失败 • PARTIALLY_SUCCESSFUL: 批次部分成功 • CANCELLED: 已取消 • SUCCESSFUL: 刷新成功 示例值: WAITING
RefreshBatchRelatedInstanceSet	Array of RefreshBatchRelatedInstance	刷新批次关联实例列表。 示例值: [{"InstanceId": "ins-m30w46cu", "InstanceStatusMessage": "success", "LastActivityId": "asa-dayrcku8", "InstanceStatus": "SUCCESSFUL"}]
StartTime	Timestamp ISO8601	刷新批次开始时间。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-08-31T11:46:40Z
EndTime	Timestamp ISO8601	刷新批次结束时间。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-08-31T11:46:40Z

RefreshBatchRelatedInstance

刷新批次关联实例, 包含单个实例的刷新活动状态、对应伸缩活动等信息。

被如下接口引用：DescribeRefreshActivities。

名称	类型	描述
InstanceId	String	实例 ID。 示例值：ins-d73gg3yt
InstanceStatus	String	刷新实例状态。如果在刷新时实例被移出或销毁，状态会更新为 NOT_FOUND。取值如下： <ul style="list-style-type: none"> • WAITING：待刷新 • INIT：初始化中 • RUNNING：刷新中 • FAILED：刷新失败 • CANCELLED：已取消 • SUCCESSFUL：刷新成功 • NOT_FOUND：实例不存在 示例值：WAITING
LastActivityId	String	实例刷新中最近一次伸缩活动 ID，可通过 DescribeAutoScalingActivities 接口查询。 需注意伸缩活动与实例刷新活动不同，一次实例刷新活动可能包括多次伸缩活动。 示例值：asa-7y5rey56
InstanceStatusMessage	String	实例刷新状态信息。 示例值：success

RefreshSettings

实例刷新设置。

被如下接口引用：DescribeRefreshActivities, RollbackInstanceRefresh, StartInstanceRefresh。

名称	类型	必选	描述
RollingUpdateSettings	RollingUpdateSettings	是	滚动更新设置参数。RefreshMode 为滚动更新该参数必须填写。 示例值：{"BatchNumber": 1, "MaxSurge": 1, "BatchPause": "AUTOMATIC"}
CheckInstanceTargetHealth	Boolean	否	实例后端服务健康状态检查，默认为 FALSE。仅针对绑定应用型负载均衡的伸缩组生效，开启该检查后，如刷新后实例未通过检查，负载均衡器端口权重始终为 0，且标记为刷新失败。取值范围如下： <ul style="list-style-type: none"> • TRUE：开启检查 • FALSE：不开启检查 示例值：TRUE

RelatedInstance

与本次伸缩活动相关的实例信息。

被如下接口引用：DescribeAutoScalingActivities, DescribeAutoScalingGroupLastActivities。

名称	类型	描述
InstanceId	String	实例 ID。 示例值：ins-d7y6drte
InstanceStatus	String	实例在伸缩活动中的状态。取值如下： <ul style="list-style-type: none"> INIT：初始化中 RUNNING：实例操作中 SUCCESSFUL：活动成功 FAILED：活动失败 示例值：RUNNING

RollingUpdateSettings

滚动更新设置。

被如下接口引用：DescribeRefreshActivities, RollbackInstanceRefresh, StartInstanceRefresh。

名称	类型	必选	描述
BatchNumber	Integer	是	批次数量。批次数量为大于 0 的正整数，但不能大于待刷新实例数量。 示例值：1
BatchPause	String	否	批次间暂停策略。默认值为 Automatic，取值范围如下： <ul style="list-style-type: none"> FIRST_BATCH_PAUSE：第一批次更新完成后暂停 BATCH_INTERVAL_PAUSE：批次间暂停 AUTOMATIC：不暂停 示例值：AUTOMATIC
MaxSurge	Integer	否	最大额外数量。设置该参数后，在滚动更新开始前根据启动配置创建一批按量计费的额外实例，滚动更新完成后销毁额外实例。 该参数用于保证滚动更新过程中可用实例的数量，最大额外数量不能超过滚动更新单个批次的刷新实例数。回滚流程暂不支持该参数。 示例值：1
FailProcess	String	否	失败处理策略。默认值为 AUTO_PAUSE，取值范围如下： <ul style="list-style-type: none"> AUTO_PAUSE：刷新失败后暂停 AUTO_ROLLBACK：刷新失败后回滚。回滚时每批次回滚一台实例，CheckInstanceTargetHealth 参数值与原刷新活动一致。MaxSurge 参数引入的扩缩容流程失败无需回滚，会用取消动作代替回滚 AUTO_CANCEL：刷新失败后取消 示例值：AUTO_PAUSE

RunAutomationServiceEnabled

描述了“自动化助手”服务相关的信息

被如下接口引用：CreateLaunchConfiguration, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
Enabled	Boolean	否	是否开启 自动化助手 服务。取值范围： <ul style="list-style-type: none"> TRUE：表示开启自动化助手服务 FALSE：表示不开启自动化助手服务 注意：此字段可能返回 null，表示取不到有效值。 示例值：True

RunMonitorServiceEnabled

描述了“可观测平台（原名云监控）”服务相关的信息。

被如下接口引用：CreateLaunchConfiguration, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
Enabled	Boolean	否	是否开启 可观测平台（原名云监控） 服务。取值范围： <ul style="list-style-type: none"> TRUE：表示开启云监控服务 FALSE：表示不开启云监控服务 默认取值：TRUE。 示例值：{"Enabled": true}

RunSecurityServiceEnabled

描述了“云安全”服务相关的信息

被如下接口引用: CreateLaunchConfiguration, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
Enabled	Boolean	否	是否开启云安全服务。取值范围： <ul style="list-style-type: none"> TRUE：表示开启云安全服务 FALSE：表示不开启云安全服务 默认取值：TRUE。 示例值：{"Enabled": true}

ScalingPolicy

告警触发策略。

被如下接口引用: DescribeScalingPolicies。

名称	类型	描述
AutoScalingGroupId	String	伸缩组ID。 示例值: asg-12345678
AutoScalingPolicyId	String	告警触发策略ID。 示例值: asp-12345678
ScalingPolicyType	String	告警触发策略类型。取值： <ul style="list-style-type: none"> SIMPLE：简单策略 TARGET_TRACKING：目标追踪策略 示例值: SIMPLE
ScalingPolicyName	String	告警触发策略名称。 示例值: test-asp
AdjustmentType	String	告警触发后，期望实例数修改方式，仅适用于简单策略。取值范围： <ul style="list-style-type: none"> CHANGE_IN_CAPACITY：增加或减少若干期望实例数 EXACT_CAPACITY：调整至指定期望实例数 PERCENT_CHANGE_IN_CAPACITY：按百分比调整期望实例数 示例值: CHANGE_IN_CAPACITY
AdjustmentValue	Integer	告警触发后，期望实例数的调整值，仅适用于简单策略。 示例值: 2
Cooldown	Integer	冷却时间，单位为秒，仅适用于简单策略。取值范围 [0,3600]，默认冷却时间300秒。 示例值: 60
MetricAlarm	MetricAlarm	简单告警触发策略告警监控指标，仅适用于简单策略。 示例值: {"ComparisonOperator": "GREATER_THAN", "ContinuousTime": 1, "MetricName": "CPU_UTILIZATION", "Period": 60, "Statistic": "AVERAGE", "Threshold": 15}
PredefinedMetricType	String	预定义监控项，仅适用于目标追踪策略。取值范围： <ul style="list-style-type: none"> ASG_AVG_CPU_UTILIZATION：平均CPU使用率 ASG_AVG_LAN_TRAFFIC_OUT：平均内网出带宽 ASG_AVG_LAN_TRAFFIC_IN：平均内网入带宽 ASG_AVG_WAN_TRAFFIC_OUT：平均外网出带宽 ASG_AVG_WAN_TRAFFIC_IN：平均外网入带宽 注意：此字段可能返回 null，表示取不到有效值。 示例值: ASG_AVG_CPU_UTILIZATION
TargetValue	Integer	目标值，仅适用于目标追踪策略。 <ul style="list-style-type: none"> ASG_AVG_CPU_UTILIZATION: [1, 100)，单位：% ASG_AVG_LAN_TRAFFIC_OUT: >0，单位：Mbps

名称	类型	描述
		<ul style="list-style-type: none"> ASG_AVG_LAN_TRAFFIC_IN: >0, 单位: Mbps ASG_AVG_WAN_TRAFFIC_OUT: >0, 单位: Mbps ASG_AVG_WAN_TRAFFIC_IN: >0, 单位: Mbps <p>注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 10</p>
EstimatedInstanceWarmup	Integer	实例预热时间, 单位为秒, 仅适用于目标追踪策略。取值范围为0-3600。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 300
DisableScaleIn	Boolean	是否禁用缩容, 仅适用于目标追踪策略。取值范围: <ul style="list-style-type: none"> true: 目标追踪策略仅触发扩容 false: 目标追踪策略触发扩容和缩容 <p>注意: 此字段可能返回 null, 表示取不到有效值。 示例值: false</p>
MetricAlarms	Array of MetricAlarm	告警监控指标列表, 仅适用于目标追踪策略。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"ComparisonOperator": "GREATER_THAN", "Period": 300, "ContinuousTime": 1, "Threshold": 35, "Statistic": "AVERAGE", "PreciseThreshold": 35.0, "MetricName": "CPU_UTILIZATION"}
NotificationUserGroupIds	Array of String	通知组ID, 即为用户组ID集合。 示例值: ["12001"]

ScheduledAction

描述定时任务的信息

被如下接口引用: DescribeScheduledActions。

名称	类型	描述
ScheduledActionId	String	定时任务ID。 示例值: asst-jf898dps
ScheduledActionName	String	定时任务名称。 示例值: as-sche-action-0
AutoScalingGroupID	String	定时任务所在伸缩组ID。 示例值: asg-keqt9eg1
StartTime	Timestamp ISO8601	定时任务的开始时间。取值为北京时间 (UTC+8), 按照ISO8601标准, 格式: YYYY-MM-DDThh:mm:ss+08:00。 示例值: 2022-02-22T16:00:00+08:00
Recurrence	String	定时任务的重复方式。定时任务中的 Recurrence 参数限制为5个字段, 由空格分开, 结构为: 分, 小时, 日期, 月份, 星期。 示例值: 0 23 */1 * *
EndTime	Timestamp ISO8601	定时任务的结束时间。取值为北京时间 (UTC+8), 按照ISO8601标准, 格式: YYYY-MM-DDThh:mm:ss+08:00。 示例值: 2023-02-23T16:00:00+08:00
MaxSize	Integer	定时任务设置的最大实例数。 示例值: 20
DesiredCapacity	Integer	定时任务设置的期望实例数。 示例值: 5

名称	类型	描述
MinSize	Integer	定时任务设置的最小实例数。 示例值：1
CreatedTime	Timestamp ISO8601	定时任务的创建时间。取值为标准UTC时间，按照ISO8601标准，格式：YYYY-MM-DDThh:mm:ssZ。 示例值：2022-02-21T02:19:52Z
ScheduledType	String	定时任务的执行类型。取值范围： <ul style="list-style-type: none"> CRONTAB：代表定时任务为重复执行。 ONCE：代表定时任务为单次执行。 示例值：CRONTAB

ServiceSettings

服务设置

被如下接口引用：CreateAutoScalingGroup, DescribeAutoScalingGroups, ModifyAutoScalingGroup。

名称	类型	必选	描述
ReplaceMonitorUnhealthy	Boolean	否	开启监控不健康替换服务。若开启则对于云监控标记为不健康的实例，弹性伸缩服务会进行替换。若不指定该参数，则默认为 False。 示例值：False
ScalingMode	String	否	取值范围： CLASSIC_SCALING：经典方式，使用创建、销毁实例来实现扩缩容； WAKE_UP_STOPPED_SCALING：扩容优先开机。扩容时优先对已关机的实例执行开机操作，若开机后实例数仍低于期望实例数，则创建实例，扩容仍采用销毁实例的方式。用户可以使用StopAutoScalingInstances接口来关闭伸缩组内的实例。监控告警触发的扩容仍将创建实例 默认取值：CLASSIC_SCALING 示例值：CLASSIC_SCALING
ReplaceLoadBalancerUnhealthy	Boolean	否	开启负载均衡不健康替换服务。若开启则对于负载均衡健康检查判断不健康的实例，弹性伸缩服务会进行替换。若不指定该参数，则默认为 False。 示例值：False
ReplaceMode	String	否	不健康替换服务的替换模式。取值范围： RECREATE：重建实例替代原有不健康实例； RESET：对原有不健康实例进行重装系统操作，可保持数据盘、内网IP、实例id等信息不发生变化，实例登录设置、主机名、增强服务和 UserData 与当前启动配置保持一致。 默认取值：RECREATE 示例值：RECREATE
AutoUpdateInstanceTags	Boolean	否	自动更新实例标签。默认取值为 False，配置后如伸缩组标签发生更新，会同步更新（同步更新仅支持新增、修改标签，暂不支持删除标签）伸缩组内运行中状态实例的标签，同步更新非立即生效，存在一定延迟。 示例值：False
DesiredCapacitySyncWithMaxMinSize	Boolean	否	期望实例数同步最大最小值。默认值为 False。该参数仅对修改伸缩组接口未传入期望数的场景生效。 <ul style="list-style-type: none"> True：修改最大值或最小值时，如与当前期望数存在冲突，则同步调整期望数。例如修改时传入最小值 2，当前期望数为 1，则同步调整期望数为 2。 False：修改最大值或最小值时，如与当前期望数存在冲突，报错提示不允许修改。 示例值：False

SpotMarketOptions

竞价相关选项

被如下接口引用: CreateLaunchConfiguration, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
MaxPrice	String	是	竞价出价, 例如 “1.05” 示例值: 1.05
SpotInstanceType	String	否	竞价请求类型, 当前仅支持类型: one-time, 默认值为one-time 示例值: one-time

SpotMixedAllocationPolicy

竞价混合模式下, 各计费类型实例的分配策略。包括按量计费实例和竞价计费实例。

被如下接口引用: CreateAutoScalingGroup, DescribeAutoScalingGroups, ModifyAutoScalingGroup。

名称	类型	必选	描述
BaseCapacity	Integer	否	混合模式下, 基础容量的大小, 基础容量部分固定为按量计费实例。默认值 0, 最大不可超过伸缩组的最大实例数。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
OnDemandPercentageAboveBaseCapacity	Integer	否	超出基础容量部分, 按量计费实例所占的比例。取值范围 [0, 100], 0 代表超出基础容量的部分仅生产竞价实例, 100 代表仅生产按量实例, 默认值为 70。按百分比计算按量实例数时, 向上取整。 比如, 总期望实例数取 3, 基础容量取 1, 超基础部分按量百分比取 1, 则最终按量 2 台 (1 台来自基础容量, 1 台按百分比向上取整得到), 竞价 1 台。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
SpotAllocationStrategy	String	否	混合模式下, 竞价实例的分配策略。取值包括 COST_OPTIMIZED 和 CAPACITY_OPTIMIZED, 默认取 COST_OPTIMIZED。 <ul style="list-style-type: none"> COST_OPTIMIZED, 成本优化策略。对于启动配置内的所有机型, 按照各机型在各可用区的每核单价由小到大依次尝试。优先尝试购买每核单价最便宜的, 如果购买失败则尝试购买次便宜的, 以此类推。 CAPACITY_OPTIMIZED, 容量优化策略。对于启动配置内的所有机型, 按照各机型在各可用区的库存情况由大到小依次尝试。优先尝试购买剩余库存最大的机型, 这样可尽量降低竞价实例被动回收的发生概率。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: COST_OPTIMIZED
CompensateWithBaseInstance	Boolean	否	按量实例替补功能。取值范围: <ul style="list-style-type: none"> TRUE, 开启该功能, 当所有竞价机型因库存不足等原因全部购买失败后, 尝试购买按量实例。 FALSE, 不开启该功能, 伸缩组在需要扩容竞价实例时仅尝试所配置的竞价机型。 默认取值: TRUE。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: false

SystemDisk

启动配置的系统盘配置信息。若不指定该参数, 则按照系统默认值进行分配。

被如下接口引用: CreateLaunchConfiguration, DescribeLaunchConfigurations, ModifyLaunchConfigurationAttributes, UpgradeLaunchConfiguration。

名称	类型	必选	描述
DiskType	String	否	系统盘类型。系统盘类型限制详见 云硬盘类型 。取值范围 <ul style="list-style-type: none"> • LOCAL_BASIC: 本地硬盘 • LOCAL_SSD: 本地SSD硬盘 • CLOUD_BASIC: 普通云硬盘 • CLOUD_PREMIUM: 高性能云硬盘 • CLOUD_SSD: SSD云硬盘 • CLOUD_BSSD: 通用型SSD云硬盘 • CLOUD_HSSD: 增强型SSD云硬盘 • CLOUD_TSSD: 极速型SSD云硬盘 • 默认取值: CLOUD_PREMIUM。 示例值: CLOUD_PREMIUM
DiskSize	Integer	否	系统盘大小, 单位: GB。默认值为 50 示例值: 50

Tag

资源类型及标签键值对

被如下接口引用: CreateAutoScalingGroup, CreateLaunchConfiguration, DescribeAutoScalingGroups, DescribeLaunchConfigurations。

名称	类型	必选	描述
Key	String	是	标签键 示例值: as-group-tag0
Value	String	是	标签值 示例值: app0
ResourceType	String	否	标签绑定的资源类型, 当前支持类型: "auto-scaling-group", "launch-configuration"。分别表示: 伸缩组资源, 启动配置资源。 示例值: auto-scaling-group

TargetAttribute

负载均衡器目标属性

被如下接口引用: AttachLoadBalancers, CreateAutoScalingGroup, DescribeAutoScalingGroups, ModifyLoadBalancerTargetAttributes, ModifyLoadBalancers。

名称	类型	必选	描述
Port	Integer	是	端口。取值范围为 [1,65535]。作为入参时, 该参数必填。 示例值: 80
Weight	Integer	是	权重。取值范围为 [0,100]。作为入参时, 该参数必填。 示例值: 10

错误码

最近更新时间：2025-06-11 01:05:04

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
ActionOffline	接口已下线。
AuthFailure.InvalidAuthorization	请求头部的 Authorization 不符合腾讯云标准。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。请在 控制台 检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误（包括参数格式、类型等错误）。
InvalidParameterValue	参数取值错误。
InvalidRequest	请求 body 的 multipart 格式错误。
IplnBlacklist	IP 地址在黑名单中。

错误码	说明
IpNotInWhitelist	IP 地址不在白名单中。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
RequestLimitExceeded.GlobalRegionUinLimitExceeded	主账号超过频率限制。
RequestLimitExceeded.IPLimitExceeded	IP 限频。
RequestLimitExceeded.UinLimitExceeded	主账号限频。
RequestSizeLimitExceeded	请求包超过限制大小。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResponseSizeLimitExceeded	返回包超过限制大小。
ServiceUnavailable	当前服务暂时不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误，用户多传未定义的参数会导致错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s) 请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
AccountQualificationRestrictions	该请求账户未通过资格审计。
CallCvmError	CVM接口调用失败。
FailedOperation.NoActivityToGenerate	未生成伸缩活动。
InternalError.CallCmqError	Cmq 接口调用失败。
InternalError.CallError	内部接口调用失败。
InternalError.CallLbError	LB 接口调用失败。
InternalError.CallMonitorError	Monitor接口调用失败。
InternalError.CallNotificationError	通知服务接口调用失败。
InternalError.CallStsError	STS 接口调用失败。
InternalError.CallTATError	TAT 接口调用失败。
InternalError.CallTagError	Tag 接口调用失败。

错误码	说明
InternalServerError.CallTvpError	Tvp接口调用失败。
InternalServerError.CallVpcError	VPC接口调用失败。
InternalServerError.CalleeError	调用其他服务异常。
InternalServerError.RequestError	内部请求错误。
InvalidImageId.NotFound	未找到该镜像。
InvalidLaunchConfiguration	无效的启动配置。
InvalidLaunchConfigurationId	启动配置ID无效。
InvalidParameter.ActionNotFound	无效的Action请求。
InvalidParameter.Conflict	参数冲突，指定的多个参数冲突，不能同时存在。
InvalidParameter.HostNameUnavailable	主机名参数不适用于该镜像。
InvalidParameter.InScenario	在特定场景下的不合法参数。
InvalidParameter.InvalidCombination	无效的参数组合。
InvalidParameter.LoadBalancerNotInAutoScalingGroup	指定的负载均衡器在当前伸缩组中没有找到。
InvalidParameter.MustOneParameter	参数缺失，两种参数之中必须指定其中一个。
InvalidParameter.ParameterDeprecated	该参数已废弃。
InvalidParameter.ParameterMustBeDeleted	部分参数存在互斥应该删掉。
InvalidParameterConflict	指定的两个参数冲突，不能同时存在。
InvalidParameterValue.AccountNotSupportBandwidthPackageId	当前账户不支持带宽包ID参数。
InvalidParameterValue.AssertDesiredCapacityFailed	声明的原有期望实例数和实际原有期望实例数不一致。
InvalidParameterValue.BaseCapacityTooLarge	指定的基础容量过大，需小于等于最大实例数。
InvalidParameterValue.BatchNumberTooLarge	批次数不能大于待刷新实例总数。
InvalidParameterValue.ClassicLb	在应当指定传统型负载均衡器的参数中，错误地指定了一个非传统型的负载均衡器。
InvalidParameterValue.ConflictNotificationTarget	通知接收端类型冲突。
InvalidParameterValue.CronExpressionIllegal	定时任务指定的Cron表达式无效。
InvalidParameterValue.CvmConfigurationError	CVM参数校验异常。
InvalidParameterValue.CvmError	CVM参数校验异常。
InvalidParameterValue.DuplicatedForwardLb	提供的应用型负载均衡器重复。
InvalidParameterValue.DuplicatedSubnet	指定的子网重复。
InvalidParameterValue.EndTimeBeforeStartTime	定时任务设置的结束时间在开始时间。
InvalidParameterValue.Filter	无效的过滤器。
InvalidParameterValue.ForwardLb	在应当指定应用型负载均衡器的参数中，错误地指定了一个非应用型的负载均衡器。
InvalidParameterValue.GroupNameDuplicated	伸缩组名称重复。
InvalidParameterValue.HostNameIllegal	主机名不合法。

错误码	说明
InvalidParameterValue.HostNameWithSuffixTooLong	带后缀的主机名称过长，超过了规定上限。
InvalidParameterValue.IPv6InternetChargeType	指定的IPv6公网带宽计费模式不合法。
InvalidParameterValue.ImageNotFound	指定的镜像不存在。
InvalidParameterValue.InstanceNameIllegal	设置的实例名称不合法。
InvalidParameterValue.InstanceNameWithSuffixTooLong	带后缀的实例名称过长，超过了规定上限。
InvalidParameterValue.InstanceTypeNotSupported	实例机型不支持。
InvalidParameterValue.InvalidActivityId	伸缩活动ID无效。
InvalidParameterValue.InvalidAutoScalingGroupId	伸缩组ID无效。
InvalidParameterValue.InvalidAutoScalingNotificationId	通知ID无效。
InvalidParameterValue.InvalidAutoScalingPolicyId	告警策略ID无效。
InvalidParameterValue.InvalidClbRegion	为CLB指定的地域不合法。
InvalidParameterValue.InvalidDisasterRecoverGroupId	指定的置放群组ID格式不正确。
InvalidParameterValue.InvalidFilter	过滤条件无效。
InvalidParameterValue.InvalidHpcClusterId	高性能计算集群ID无效。
InvalidParameterValue.InvalidImageId	镜像ID无效。
InvalidParameterValue.InvalidInstanceId	实例ID无效。
InvalidParameterValue.InvalidInstanceType	实例机型无效。
InvalidParameterValue.InvalidLaunchConfiguration	输入的启动配置无效。
InvalidParameterValue.InvalidLaunchConfigurationId	启动配置ID无效。
InvalidParameterValue.InvalidLifecycleHookId	生命周期挂钩ID无效。
InvalidParameterValue.InvalidNotificationUserGroupId	指定的通知组 ID 不是数值字符串格式。
InvalidParameterValue.InvalidScheduledActionId	定时任务ID无效。
InvalidParameterValue.InvalidScheduledActionNameIncludIllegalChar	定时任务名称包含无效字符。
InvalidParameterValue.InvalidSecurityGroupId	安全组ID无效。
InvalidParameterValue.InvalidSnapshotId	快照ID无效。
InvalidParameterValue.InvalidSubnetId	子网ID无效。
InvalidParameterValue.LaunchConfigurationNameDuplicated	启动配置名称重复。
InvalidParameterValue.LaunchConfigurationNotFound	找不到指定启动配置。
InvalidParameterValue.LbProjectInconsistent	负载均衡器项目不一致。
InvalidParameterValue.LifecycleHookNameDuplicated	生命周期挂钩名称重复。
InvalidParameterValue.LimitExceeded	取值超出限制。
InvalidParameterValue.ListenerTargetTypeNotSupported	不支持目标组类型的监听器。
InvalidParameterValue.MaxSurgeTooLarge	最大额外数量取值过大。
InvalidParameterValue.MissingBandwidthPackageId	当前账户在带宽包模式中必须填写带宽包ID参数。

错误码	说明
InvalidParameterValue.NoResourcePermission	无资源权限。
InvalidParameterValue.NotStringTypeFloat	提供的值不是浮点字符串格式。
InvalidParameterValue.OnlyVpc	账号仅支持VPC网络。
InvalidParameterValue.ProjectIdNotFound	项目ID不存在。
InvalidParameterValue.Range	取值超出指定范围。
InvalidParameterValue.ScalingPolicyNameDuplicate	告警策略名称重复。
InvalidParameterValue.ScheduledActionNameDuplicate	定时任务名称重复。
InvalidParameterValue.Size	伸缩组最大数量、最小数量、期望实例数取值不合法。
InvalidParameterValue.StartTimeBeforeCurrentTime	定时任务设置的开始时间在当前时间之前。
InvalidParameterValue.SubnetIds	子网信息不合法。
InvalidParameterValue.SubnetNotInDedicatedCluster	子网不在指定本地专用集群中。
InvalidParameterValue.TargetPortDuplicated	负载均衡器四层监听器的后端端口重复。
InvalidParameterValue.TargetTrackingScalingPolicy	不支持执行目标追踪策略。
InvalidParameterValue.ThresholdOutOfRange	指定的阈值不在有效范围。
InvalidParameterValue.TimeFormat	时间格式错误。
InvalidParameterValue.TooLong	取值过多。
InvalidParameterValue.TooShort	输入参数值的长度小于最小值。
InvalidParameterValue.UserDataFormatError	UserData格式错误。
InvalidParameterValue.UserDataSizeExceeded	UserData长度过长。
InvalidParameterValue.UserGroupIdNotFound	用户组不存在。
InvalidParameterValue.ZoneMismatchRegion	指定的可用区与地域不匹配。
InvalidPermission	账户不支持该操作。
LimitExceeded.AfterAttachLbLimitExceeded	绑定指定的负载均衡器后，伸缩组绑定的负载均衡器总数超过了最大值。
LimitExceeded.AutoScalingGroupLimitExceeded	伸缩组数量超过限制。
LimitExceeded.DesiredCapacityLimitExceeded	期望实例数超出限制。
LimitExceeded.FilterValuesTooLong	特定过滤器的值过多。
LimitExceeded.LaunchConfigurationQuotaNotEnough	启动配置配额不足。
LimitExceeded.MaxSizeLimitExceeded	最大实例数大于限制。
LimitExceeded.MinSizeLimitExceeded	最小实例数低于限制。
LimitExceeded.QuotaNotEnough	当前剩余配额不足。
LimitExceeded.ScheduledActionLimitExceeded	定时任务数量超过限制。
LimitExceeded.TargetTrackingScalingPolicy	每个伸缩组限制创建一个目标追踪策略。
MissingParameter.InScenario	在特定场景下缺少参数。
MissingParameter.InstanceMarketOptions	竞价计费类型缺少对应的 InstanceMarketOptions 参数。

错误码	说明
ResourceInUse.ActivityInProgress	伸缩组正在执行伸缩活动。
ResourceInUse.AutoScalingGroupNotActive	伸缩组处于禁用状态。
ResourceInUse.InstanceInGroup	伸缩组内尚有正常实例。
ResourceInUse.LaunchConfigurationIdInUse	指定的启动配置仍在伸缩组中使用。
ResourceInsufficient.AutoScalingGroupAboveMaxSize	超过伸缩组最大实例数。
ResourceInsufficient.AutoScalingGroupBelowMinSize	少于伸缩组最小实例数。
ResourceInsufficient.InServiceInstanceAboveMaxSize	伸缩组内实例数超过最大实例数。
ResourceInsufficient.InServiceInstanceBelowMinSize	伸缩组内实例数低于最小实例数。
ResourceNotFound.AutoScalingGroupIdNotFound	伸缩组不存在。
ResourceNotFound.AutoScalingGroupNotFound	伸缩组不存在。
ResourceNotFound.AutoScalingNotificationNotFound	通知不存在。
ResourceNotFound.BandwidthPackageIdNotFound	指定的带宽包ID不存在。
ResourceNotFound.CmqQueueNotFound	指定的 CMQ queue 不存在。
ResourceNotFound.CommandNotFound	命令不存在。
ResourceNotFound.DisasterRecoverGroupNotFound	指定的置放群组ID不存在。
ResourceNotFound.InstancesNotFound	指定的实例不存在。
ResourceNotFound.InstancesNotInAutoScalingGroup	目标实例不在伸缩组内。
ResourceNotFound.LaunchConfigurationIdNotFound	指定的启动配置不存在。
ResourceNotFound.LifecycleHookInstanceNotFound	生命周期挂钩对应实例不存在。
ResourceNotFound.LifecycleHookNotFound	无法找到指定生命周期挂钩。
ResourceNotFound.LifecycleHookTokenNotFound	指定的生命周期挂钩Token不存在。
ResourceNotFound.ListenerNotFound	指定的Listener不存在。
ResourceNotFound.LoadBalancerNotFound	找不到指定负载均衡器。
ResourceNotFound.LoadBalancerNotInAutoScalingGroup	指定的负载均衡器在当前伸缩组中没有找到。
ResourceNotFound.LocationNotFound	指定的Location不存在。
ResourceNotFound.RefreshActivityNotFound	指定的实例刷新活动不存在。
ResourceNotFound.ScalingPolicyNotFound	告警策略不存在。
ResourceNotFound.ScheduledActionNotFound	指定的定时任务不存在。
ResourceNotFound.TDMQCMQQueueNotFound	TDMQ-CMQ 队列不存在。
ResourceNotFound.TDMQCMQTopicNotFound	TDMQ-CMQ 主题不存在。
ResourceUnavailable.AutoScalingGroupAbnormalStatus	伸缩组状态异常。
ResourceUnavailable.AutoScalingGroupDisabled	伸缩组被停用。
ResourceUnavailable.AutoScalingGroupInActivity	伸缩组正在活动中。
ResourceUnavailable.AutoScalingGroupInRefreshActivity	伸缩组处于另一个实例刷新活动当中。

错误码	说明
ResourceUnavailable.CmqTopicHasNoSubscriber	指定的 CMQ Topic 无订阅者。
ResourceUnavailable.CvmVpcInconsistent	实例和伸缩组Vpc不一致。
ResourceUnavailable.ForbiddenModifyVpc	绑定负载均衡器的伸缩组禁止修改Vpc。
ResourceUnavailable.InquiryPriceResetInstanceFailed	重装实例询价失败，新镜像与该实例其他参数存在冲突或新镜像不存在。
ResourceUnavailable.InstanceCannotAttach	实例无法添加至伸缩组。
ResourceUnavailable.InstanceInOperation	指定的实例正在活动中。
ResourceUnavailable.InstanceNotSupportStopCharging	实例不支持关机不收费。
ResourceUnavailable.InstancesAlreadyInAutoScalingGroup	实例已存在于伸缩组中。
ResourceUnavailable.LaunchConfigurationStatusAbnormal	启动配置状态异常。
ResourceUnavailable.LbBackendRegionInconsistent	CLB实例的后端地域与AS服务所在地域不一致。
ResourceUnavailable.LbProjectInconsistent	负载均衡器项目不一致。
ResourceUnavailable.LbVpcInconsistent	负载均衡器VPC与伸缩组不一致。
ResourceUnavailable.LifecycleActionResultHasSet	生命周期动作已经被设置。
ResourceUnavailable.LoadBalancerInOperation	LB 在指定的伸缩组内处于活动中。
ResourceUnavailable.NoInstanceCanRefresh	伸缩组内不存在运行中状态的实例，无法进行实例刷新。
ResourceUnavailable.NoInstanceCanRollback	伸缩组中不存在可回滚的实例。
ResourceUnavailable.ProjectInconsistent	项目不一致。
ResourceUnavailable.RefreshActivityCanNotRollback	当前刷新活动为成功状态，且不是最近一次执行的刷新活动，无法回滚。
ResourceUnavailable.RefreshActivityStatusConflictWithOperation	实例刷新活动状态与当前操作冲突。
ResourceUnavailable.RollbackTypeActivityCanNotRollbackAgain	回滚类型的实例刷新活动无法再次回滚。
ResourceUnavailable.StoppedInstanceNotAllowAttach	关机实例不允许添加到伸缩组。
ResourceUnavailable.TDMQCMQTopicHasNoSubscriber	TDMQ-CMQ 主题无订阅者。
ResourceUnavailable.ZoneUnavailable	指定的可用区不可用。
UnauthorizedOperation.AutoScalingRoleUnauthorized	未授权弹性伸缩预设服务角色AS_QCSRole，请前往弹性伸缩控制台进行授权。