Amazon EC2 Auto Scaling

用户指南

Amazon EC2 Auto Scaling: 用户指南

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆或者贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其它商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Amazon Web Services 文档中描述的 Amazon Web Services 服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅 中国的 Amazon Web Services 服务入门 (PDF)。

Table of Contents

什么是 Amazon EC2 Auto Scaling？ 1
Auto Scaling 组件 1
Amazon EC2 Auto Scaling 的定价 1
开始使用 2
相关服务 2
使用 Auto Scaling 组 2
Auto Scaling 优势 2
示例：覆盖可变需求 3
示例：Web 应用程序架构 4
示例：在可用区之间分配实例 5
实例生命周期 7
扩展 7
已投入使用的实例 8
缩小 8
附加实例 9
分离实例 9
生命周期钩子 9
进入和退出备用状态 9
配额 10
设置 12
准备使用 Amazon EC2 12
准备使用 Amazon CLI 12
开始使用 13
演练摘要 13
准备演练 14
步骤 1：创建启动模板 14
步骤 2：创建单个实例 Auto Scaling 组 15
步骤 3：验证您的 Auto Scaling 组 16
步骤 4：终止 Auto Scaling 组中的实例 16
步骤 5：后续步骤 17
步骤 6：清除 17
启动模板 19
权限 19
为 Auto Scaling 组创建启动模板 20
创建启动模板（控制台） 20
从现有实例创建启动模板（控制台） 26
其他信息 26
限制 26
将启动配置复制到启动模板 26
使用启动模板替换启动配置 27
请求竞价型实例 28
使用启动模板的 Amazon CLI 示例 29
示例用法 29
创建基本的启动模板 30
指定在启动时标记实例的标签 30
指定要传递到实例的 IAM 角色 30
分配公有 IP 地址 31
指定用于在启动时配置实例的用户数据脚本 31
指定块储存设备映射 31
指定专属主机以从外部供应商获得软件许可证 31
指定现有网络接口 31
创建多个网络接口 32
管理启动模板 32
更新 Auto Scaling 组以使用启动模板 34

使用 Systems Manager 参数代替 AMI ID 35
限制 38
启动配置 39
创建启动配置 39
创建启动配置（控制台） 40
创建启动配置（Amazon CLI） 41
配置 IMDS 41
使用 EC2 实例创建启动配置 42
使用 EC2 实例创建启动配置 43
从实例创建启动配置以及覆盖块储存设备 (Amazon CLI) 44
创建启动配置和覆盖实例类型 (Amazon CLI) 45
更改启动配置 46
配置实例租赁 46
自动扩缩组 49
使用多种实例类型和购买选项 50
分配策略 50
竞价型实例的最佳实践 52
控制按需型实例的比例 52
先决条件 54
创建包括竞价型实例和按需型实例的 Auto Scaling 组（控制台） 54
创建包括竞价型实例和按需型实例的自动扩缩组（Amazon CLI） 56
验证您的自动扩缩组配置和已启动的实例 (Amazon CLI) 63
配置覆盖 63
示例：使用基于属性的实例类型选择创建组 72
注意事项 73
先决条件 74
使用基于属性的实例类型选择 74
限制 78
使用启动模板创建组 78
使用启动配置创建组 80
使用 EC2 实例创建组 81
先决条件 82
从 EC2 实例创建 Auto Scaling 组 (控制台) 82
从 EC2 实例创建 Auto Scaling 组 (Amazon CLI) 83
使用 EC2 启动向导创建组 85
创建启动模板 86
创建 自动扩缩组 86
后续步骤 87
对组和实例进行标记 87
标签命名和使用限制 88
EC2 实例标签生命周期 88
标记 Auto Scaling 组 88
删除标签 90
安全性标签 91
控制对标签的访问 91
使用标签筛选 Auto Scaling 组 92
替换实例 94
基于实例刷新替换实例 95
基于最大实例生命周期替换实例 115
删除 Auto Scaling 基础设施 117
删除 Auto Scaling 组 118
（可选）删除启动配置 118
（可选）删除启动模板 119
（可选）删除负载均衡器和目标组 119
（可选）删除 CloudWatch 警报 120
扩展组 121
扩展选项 121

设置容量限制。 122
保持固定数量的实例 123
手动扩展 123
更改 Auto Scaling 组的大小（控制台） 123
更改 Auto Scaling 组的大小（Amazon CLI） 124
将 EC2 实例附上 Auto Scaling 组。 125
从 Auto Scaling 组中分离 EC2 实例 129
动态扩展 132
动态扩缩策略的工作方式 132
多个动态扩缩策略 133
目标跟踪扩缩策略 133
步进和简单扩展策略 141
设置预热或冷却的原定设置值 149
根据 Amazon SQS 进行扩展 155
验证扩缩活动 160
禁用扩缩策略 161
删除扩展策略 163
Amazon CLI 扩展策略示例 164
预测式扩展 166
预测式扩展的工作方式 166
最佳实践 167
创建预测式扩展策略（控制台） 167
创建预测式扩展策略 (Amazon CLI) 170
限制 172
支持的区域 172
评估预测性扩展策略 172
覆盖预测 177
使用自定义指标 180
计划的扩展 186
注意事项 187
定期安排 187
创建和管理计划操作（控制台） 187
创建和管理计划操作 (Amazon CLI) 189
限制 191
生命周期钩子 191
生命周期钩子的注意事项和限制 191
生命周期钩子可用性 192
示例 193
生命周期钩子的工作方式 193
做好准备添加生命周期钩子 194
添加生命周期钩子 199
完成生命周期操作 201
教程：配置用户数据以通过实例元数据检索目标生命周期状态 202
教程：配置调用 Lambda 函数的生命周期钩子 207
暖池 212
核心概念 213
先决条件 214
创建暖池 215
更新暖池 215
删除暖池 215
限制 216
使用生命周期钩子 216
查看运行状况检查状态 218
使用暖池的 Amazon CLI 示例 220
控制实例终止 222
终止策略方案 222
使用终止策略 224

了解使用 Lambda 创建自定义终止策略。 227
实例横向缩减保护 230
临时移除实例 232
备用状态的工作方式 233
处于备用状态的实例的运行状况 233
使用移除删除实例（控制台） 234
临时移除实例 (Amazon CLI) 234
暂停和恢复进程 236
进程的类型 236
注意事项 237
暂停和恢复扩展进程（控制台） 239
暂停和恢复进程 (Amazon CLI) 239
监控 241
检查实例运行状况 241
运行状况检查类型 242
Amazon EC2 运行状况检查 242
Elastic Load Balancing 运行状况检查 243
自定义运行状况检测 244
替换运行不正常的实例 244
Amazon EC2 Auto Scaling 如何尽可能减少停机时间 245
运行状况检查注意事项 245
其他信息 246
运行状况检查宽限期 246
使用 Amazon Health Dashboard 监控 247
使用 CloudWatch 监控 GPU 248
可用指标和维度 249
启用 Auto Scaling 组指标（控制台） 252
启用 Auto Scaling 组指标 (Amazon CLI) 252
配置 Auto Scaling 实例的监控 253
在 Amazon EC2 Auto Scaling 控制台中查看监控图表 254
使用 Amazon CloudTrail 记录 API 调用 256
CloudTrail 中的 Amazon EC2 Auto Scaling 信息 257
了解 Amazon EC2 Auto Scaling 日志文件条目 257
使用 Amazon SNS 通知进行监控 258
SNS 通知 259
为 Amazon EC2 Auto Scaling 配置 Amazon SNS 通知 259
使用其他服务 262
容量再平衡 262
工作原理 262
注意事项 263
启用容量再平衡 264
添加终止生命周期钩子 267
限制 268
Amazon CloudFormation 269
Amazon EC2 Auto Scaling 和 Amazon CloudFormation 模板 269
了解有关 Amazon CloudFormation 的更多信息 269
Amazon CloudShell 269
Compute Optimizer 270
限制 270
结果 270
查看建议 271
评估建议时的注意事项 271
弹性负载均衡 272
Elastic Load Balancing 类型 272
先决条件 273
附加负载均衡器 274
了解附加状态 275

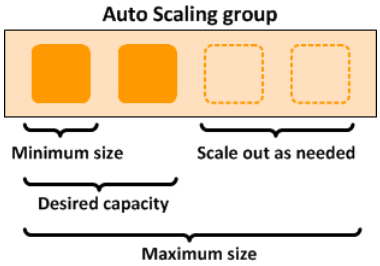
从 Amazon EC2 Auto Scaling 控制台配置负载均衡器。 276
添加 Elastic Load Balancing 运行状况检查 276
添加可用区 277
与 Elastic Load Balancing 配合使用的 Amazon CLI 示例 279
教程：设置具有扩展和负载均衡功能的应用程序 282
EventBridge 287
Amazon EC2 Auto Scaling 事件参考 288
暖池活动类型和模式 293
创建 EventBridge 规则 297
Amazon VPC 300
EC2-Classic 301
默认 VPC 301
非默认 VPC 302
选择 VPC 子网时的注意事项 302
VPC 中的 IP 寻址 302
VPC 中的网络接口 302
实例部署租期 303
Amazon Outposts 303
用于了解 VPC 的更多资源 303
安全性 304
数据保护 304
使用 Amazon KMS keys 对 Amazon EBS 卷进行加密 305
Identity and Access Management 305
访问控制 305
Amazon EC2 Auto Scaling 如何与 IAM 结合使用 306
托管式策略 312
服务相关角色 314
基于身份的策略示例 319
防止跨服务混淆代理 327
启动模板支持 328
在 Amazon EC2 实例上运行的应用程序的 IAM 角色 331
与加密卷结合使用时的 Amazon KMS 密钥策略 333
合规性验证 337
PCI DSS 合规性 337
故障恢复能力 338
基础设施安全性 338
使用 VPC 终端节点建立私有连接 339
创建接口 VPC 终端节点 339
创建 VPC 终端节点策略 339
问题排查 341
检索错误消息 341
其他故障排除资源 342
实例启动失败 343
当前不支持请求的配置。 343
安全组 <该安全组的名称> 不存在。启动 EC2 实例失败。 344
密钥对 <与您的 EC2 实例相关联的密钥对> 不存在。启动 EC2 实例失败。 344
不再支持请求的可用区。请重新尝试请求... 344
请求的实例类型（<实例类型>）在请求的可用区（<实例可用区>）中不受支持... 344
您的竞价请求价格 0.015 低于要求的最低竞价请求履行价格 0.0735... 345
设备名称 <device name> 无效/设备名称上传无效。启动 EC2 实例失败。 345
用于参数 virtualName 的值（<与实例存储设备相关联的名称>）无效... 345
实例存储 AMI 不支持 EBS 块储存设备映射。 345
置放群组可能无法与类型为“m1.large”的实例一起使用。启动 EC2 实例失败。 346
Client.InternalError：客户端在启动时出错。 346
我们目前在您请求的可用区中没有足够的 <实例类型> 容量。启动 EC2 实例失败。 346
没有与您的请求匹配的竞价容量。启动 EC2 实例失败。 347
已运行 <实例数量> 个实例。启动 EC2 实例失败。 347

AMI 问题 347
AMI ID <您的 AMI 的 ID> 不存在。启动 EC2 实例失败。 348
AMI <AMI ID> 正在等待，无法运行。启动 EC2 实例失败。 348
用于参数 virtualName 的值 (<ami ID>) 无效。 348
请求的实例类型的架构 (i386) 与用于 ami-6622f00f (x86_64) 的清单中的架构不匹配。启动 EC2 实例失败。 348
负载均衡器问题 348
一个或多个目标组未找到。验证负载均衡器配置失败。 349
找不到负载均衡器 <your load balancer>。验证负载均衡器配置失败。 349
名为 <负载均衡器名称> 的活动负载均衡器不存在。更新负载均衡器配置失败。 349
EC2 实例 <实例 ID> 不在 VPC 内。更新负载均衡器配置失败。 349
EC2 实例 <实例 ID> 在 VPC 中。更新负载均衡器配置失败。 350
启动模板问题 350
您必须使用有效的完整启动模板（无效值） 350
您没有权限使用启动模板（权限不足） 350
运行状况检查 351
实例因 EC2 实例状态检查失败而停止使用 351
实例因 EC2 计划重新启动而停止使用 352
实例因 EC2 运行状况检查而停止服务指示它已终止或停止 352
实例因 ELB 系统运行状况检查失败而停止服务 353
资源 354
文档历史记录 355

什么是 Amazon EC2 Auto Scaling？

Amazon EC2 Auto Scaling 帮助您确保具有正确数量的 Amazon EC2 实例以处理应用程序负载。您可创建 EC2 实例的集合，称为 Auto Scaling 组。您可以指定每个 Auto Scaling 组中最少的实例数量，Amazon EC2 Auto Scaling 会确保您的组中的实例永远不会低于这个数量。您可以指定每个 Auto Scaling 组中最大的实例数量，Amazon EC2 Auto Scaling 会确保您的组中的实例永远不会高于这个数量。如果您在创建组的时候或在创建组之后的任何时候指定了所需容量，Amazon EC2 Auto Scaling 会确保您的组一直具有此数量的实例。如果您指定了扩展策略，则 Amazon EC2 Auto Scaling 可以在您的应用程序的需求增加或降低时启动或终止实例。

例如，以下 Auto Scaling 组的最小大小为 1 个实例，所需的容量为 2 个实例，最大大小为 4 个实例。您制定的扩展策略是按照您指定的条件，在最大最小实例数范围内调整实例的数量。



有关 Amazon EC2 Auto Scaling 优势的更多信息，请参阅 Amazon EC2 Auto Scaling 优势 (p. 2)。

Auto Scaling 组件

下表描述 Amazon EC2 Auto Scaling 的关键组件。

	组 您的 EC2 实例整理到组 中，从而当作一个逻辑单位进行扩展和管理。当您创建一个组时，您可以指定其中 EC2 实例的最小数量、最大数量以及所需数量。有关更多信息，请参阅自动扩缩组 (p. 49)。
	配置模板 您的群组使用启动模板或启动配置（不推荐，提供较少的功能）作为其 EC2 实例的配置模板。您可以为实例指定一些信息，例如，AMI ID、实例类型、密钥对、安全组和块储存设备映射。有关更多信息，请参阅 启动模板 (p. 19) 和 启动配置 (p. 39)：
	扩展选项 Amazon EC2 Auto Scaling 提供了多种方法来扩展 Auto Scaling 组。例如，您可以将组配置为在发生指定条件时（动态扩展）或根据时间表进行扩展。有关更多信息，请参阅 扩展选项 (p. 121)。

Amazon EC2 Auto Scaling 的定价

Amazon EC2 Auto Scaling 不产生额外费用，因此可方便地试用它并了解它如何使您的 Amazon 架构获益。您仅需为使用的 Amazon 资源（例如 EC2 实例、EBS 卷和 CloudWatch 警报）付费。

开始使用

若要开始使用，请学完 Amazon EC2 Auto Scaling 入门 (p. 13) 教程，以创建一个 Auto Scaling 组并了解它在该组中的实例终止时如何进行响应。

相关服务

要跨多个实例在 Auto Scaling 组中自动分配应用程序的传入流量，可使用 Elastic Load Balancing。有关更多信息，请参阅 使用 Elastic Load Balancing 跨 Auto Scaling 组中的实例分配流量。 (p. 272)。

要监控 Auto Scaling 组和实例的利用率数据，请使用 Amazon CloudWatch。有关更多信息，请参阅 监控 Auto Scaling 组和实例的 CloudWatch 指标 (p. 248)。

要为 Amazon EC2 以外其他亚马逊云科技服务的可扩展资源配置弹性伸缩，请参阅 Application Auto Scaling 用户指南。

使用 Auto Scaling 组

您可以通过下面的任何一种方式来创建、访问和管理 Auto Scaling 组：

- Amazon Web Services Management Console – 提供了可用来访问 Auto Scaling 组的 Web 界面。如果您已注册了 Amazon Web Services 账户，则可以通过登录到 Amazon Web Services Management Console，使用导航栏上的搜索框搜索 Auto Scaling groups（Auto Scaling 组），然后选择 Auto Scaling groups（Auto Scaling 组），从而访问您的 Auto Scaling 组。
- Amazon Command Line Interface（Amazon CLI）– 提供了适用于各种Amazon Web Services的命令，支持 Windows、macOS 和 Linux 等操作系统。要了解其用法，请参阅 准备使用 Amazon CLI (p. 12)。有关更多信息，请参阅 Amazon CLI 命令参考中的弹性伸缩。
- Amazon Tools for Windows PowerShell – 为在 PowerShell 环境中编写脚本的用户提供大量 Amazon 产品的相关命令。要开始使用，请参阅 Amazon Tools for Windows PowerShell 用户指南。有关更多信息，请参阅 Amazon Tools for PowerShell Cmdlet 参考。
- Amazon SDK – 提供了特定于语言的 API 操作，并简化了许多连接详细信息，例如计算签名、处理请求重试和处理错误。有关更多信息，请参阅 Amazon 软件开发工具包。
- 查询 API – 提供了您使用 HTTPS 请求调用的低级别 API 操作。使用查询 API 是访问 Amazon Web Services 的最直接方式。但它需要您的应用程序处理低级别的详细信息，例如生成哈希值以签署请求以及处理错误。有关更多信息，请参阅 Amazon EC2 Auto Scaling API 参考。
- Amazon CloudFormation – 支持使用 CloudFormation 模板创建 Auto Scaling 组。有关更多信息，请参阅 使用 Amazon CloudFormation 创建 Auto Scaling 组 (p. 269)。

要通过编程方式连接到某个Amazon Web Service，您需要使用终端节点。有关 Amazon EC2 Auto Scaling 调用的端点的信息，请参阅《中国的亚马逊云科技入门》中的中国境内的亚马逊云科技端点和 ARN。

Amazon EC2 Auto Scaling 优势

将 Amazon EC2 Auto Scaling 添加到应用程序架构是一种最大限度利用 Amazon 云的方法。当您使用 Amazon EC2 Auto Scaling 时，您的应用程序将获得以下优势：

- 提高容错能力。Amazon EC2 Auto Scaling 可以检测到实例何时运行状况不佳并终止实例，然后启动新实例以替换它。您还可以配置 Amazon EC2 Auto Scaling 以使用多个可用区。如果一个可用区变得不可用，则 Amazon EC2 Auto Scaling 可以在另一个可用区中启动实例以进行弥补。

- 提高可用性。Amazon EC2 Auto Scaling 有助于确保应用程序始终拥有合适的容量以满足当前流量需求。
- 加强成本管理。Amazon EC2 Auto Scaling 可以根据需要动态地增加或降低容量。由于您仅为使用的 EC2 实例付费，您可以在需要的时候启动实例，并在不需要的时候终止实例以节约成本。

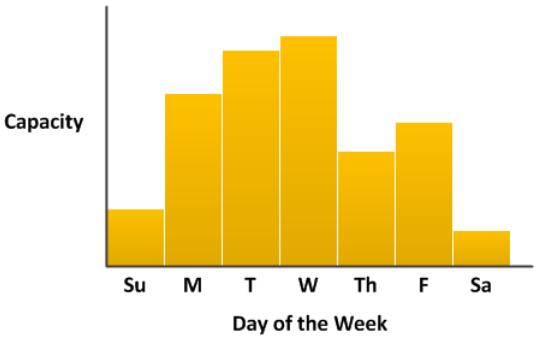
目录

- 示例：覆盖可变需求 (p. 3)
- 示例：Web 应用程序架构 (p. 4)
- 示例：在可用区之间分配实例 (p. 5)
  - 实例分配 (p. 6)
  - 再平衡活动 (p. 6)

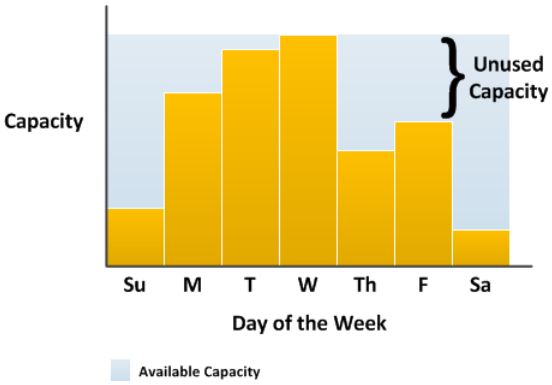
示例：覆盖可变需求

为了说明 Amazon EC2 Auto Scaling 的一些优势，请设想一个在 Amazon 上运行的基本 Web 应用程序。此应用程序允许员工搜索可用于开会的会议室。每周开始和结束时段，此应用程序的使用率最低。每周中期，有更多的员工安排会议，因此对此应用程序的需求会显著提高。

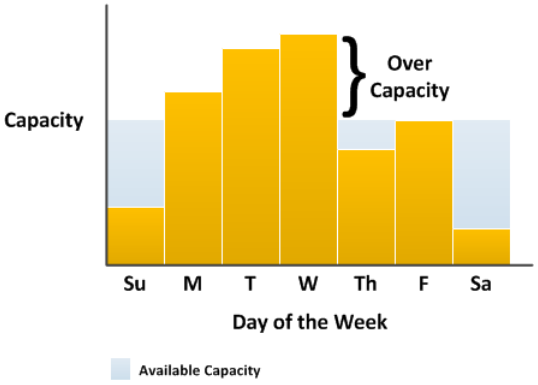
下图显示此应用程序的容量在一周中的使用情况。



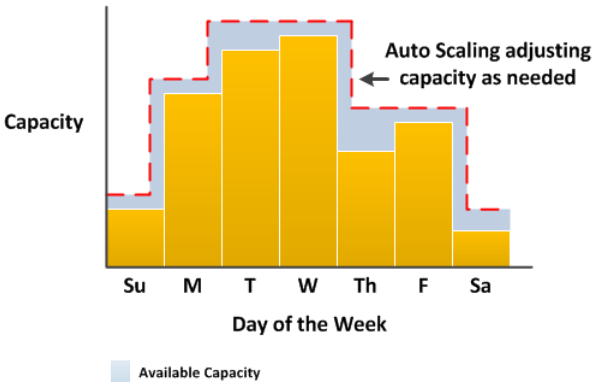
按照传统做法，可通过两种方式为这些容量变化做好规划。第一种选择是添加足够多的服务器，以便应用程序始终具有足够的容量来满足需求。但是，这种做法的缺点是应用程序在某些天并不需要这么多容量。额外容量闲置不用，并且实际上提高了使应用程序保持运行的成本。



第二种选择是采用处理应用程序平均需求所需的容量。这种做法成本更低，因为不用购买仅仅偶尔使用的设备。然而，这样做的风险是：当对应用程序的需求超过其容量时，可能造成糟糕的客户体验。

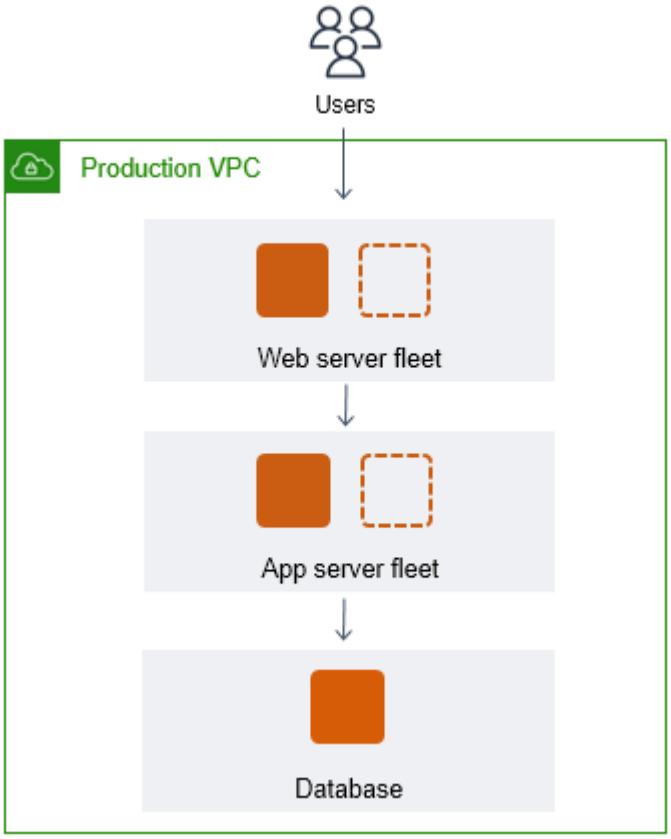


通过向此应用程序添加 Amazon EC2 Auto Scaling，您就有了第三种选择。您可以仅在需要时才向应用程序添加新实例，并在不再需要这些实例时终止它们。由于 Amazon EC2 Auto Scaling 使用 EC2 实例，因此您只需在使用时为使用的实例付费。您现在有了一个具有成本效益的架构，可在尽量减少支出的同时提供最佳客户体验。



示例：Web 应用程序架构

在常见的 Web 应用程序场景中，您同时运行应用程序的多个副本来满足客户流量。将在相同的 EC2 实例（云服务器）上托管应用程序的多个副本，每个实例都可以处理客户请求。



Amazon EC2 Auto Scaling 代表您管理 EC2 实例的启动和终止。您定义一组条件（如 Amazon CloudWatch 警报），用于确定 Auto Scaling 组何时启动或终止 EC2 实例。将 Auto Scaling 组添加到网络架构有助于提高应用程序的可用性和容错能力。

您可以根据需要创建任意数量的 Auto Scaling 组。例如，您可以为每个层创建一个 Auto Scaling 组。

要在您的 Auto Scaling 组的各实例之间分配流量，可在您的架构中引入一个负载均衡器。有关更多信息，请参阅弹性负载均衡 (p. 272)。

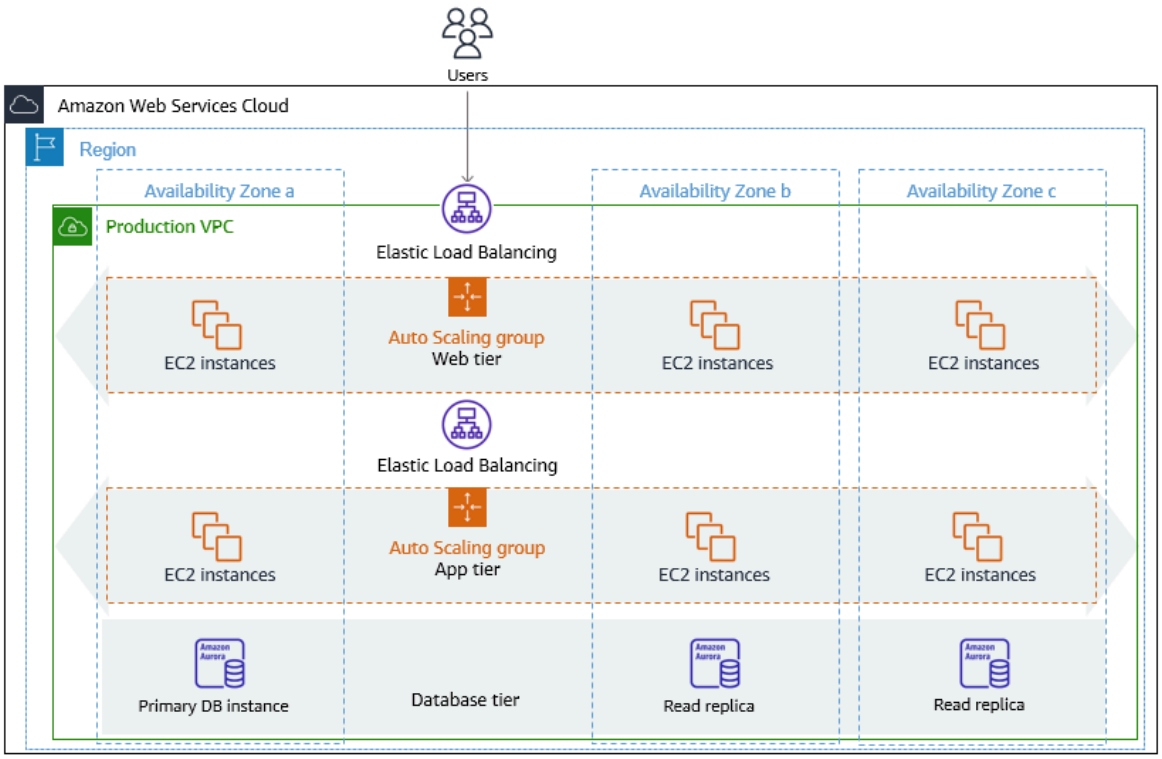
示例：在可用区之间分配实例

可用区是给定 Amazon Web Services 区域中相互隔离的站点。每个区域都有多个可用区，旨在为该区域提供高可用性。可用区相互独立，因此，设计为使用多个可用区的应用程序可以提高应用程序的可用性。有关更多信息，请参阅Amazon EC2 Auto Scaling 中的恢复功能 (p. 338)。

可用区由 Amazon Web Services 区域代码加字母标识符（例如，`us-east-1a`）来标识。如果您创建了自己的 VPC 和子网而不是使用默认 VPC，则可以在每个可用区中定义一个或多个子网。每个子网都必须完全位于一个可用区之内，不能跨越多个可用区。有关更多信息，请参阅《Amazon VPC 用户指南》中的 Amazon VPC 的工作原理。

在创建自动扩缩组时，必须选择要在其中部署该自动扩缩组的 VPC 和子网。Amazon EC2 Auto Scaling 会在您选择的子网中创建您的实例。这样，每个实例都与 Amazon EC2 Auto Scaling 选择的特定可用区相关联。当实例启动时，Amazon EC2 Auto Scaling 会尝试在可用区之间均匀分配这些实例，以实现高可用性和可靠性。

下图概括演示了跨三个可用区部署的多层级架构。



实例分配

Amazon EC2 Auto Scaling 会自动尝试在每个已启用的可用区中维持相同数量的实例。Amazon EC2 Auto Scaling 通过尝试在实例最少的可用区中启动新实例来执行此操作。如果为该可用区选择了有多个子网，则 Amazon EC2 Auto Scaling 将从该可用区中随机选择一个子网。但是，如果尝试失败，Amazon EC2 Auto Scaling 将尝试在另一可用区中启动实例，直到成功。

如果某个可用区运行状况不正常或不可用，实例在可用区之间的分布可能不再均匀。在该可用区恢复后，Amazon EC2 Auto Scaling 会自动重新平衡自动扩缩组。为此，系统将会在具有最少实例的已启用可用区中启动实例，并在其他可用区中终止实例。

再平衡活动

再平衡活动分为两类：可用区再平衡和容量再平衡。

可用区再平衡

在某些操作发生后，Auto Scaling 组可能会在不同可用区之间变得不平衡。Amazon EC2 Auto Scaling 将通过再平衡可用区来进行补偿。以下操作可能导致重新平衡活动：

- 您更改了与您的自动扩缩组关联的可用区。
- 您显式终止或分离了实例，或将实例设为待机状态，这时该组将会失衡。
- 之前没有足够容量的某个可用区已经恢复，现在具有额外的容量。
- 之前 Spot 价格超出您最高价的可用区现在的 Spot 价格低于您的最高价。

在重新平衡时，Amazon EC2 Auto Scaling 会首先启动新实例，然后再终止旧实例。这样可确保重新平衡不会影响应用程序的性能或可用性。

由于 Amazon EC2 Auto Scaling 在终止旧实例之前会尝试启动新实例，因此达到或接近指定的最大容量可能会阻碍或完全停止再平衡活动。

为避免此问题，在再平衡活动期间，系统可以暂时超出组的指定最大容量。为此，系统允许超出 10% 的边际或一个实例，以两者中最大的为准。仅在该组达到或接近最大容量并且需要重新平衡时，才会提供边际。发生这种情况的原因可能是用户请求重新分区，或者是为了补偿区域可用性问题。该超出状态仅持续重新平衡该组所需的时间（通常为几分钟）。

容量再平衡

使用竞价型实例时，您可以为您的自动扩缩组开启容量再平衡。开启此功能后，每当 Amazon EC2 报告某个竞价型实例的中断风险较高时，Amazon EC2 Auto Scaling 将会尝试启动一个竞价型实例。启动新实例后，它会终止旧实例。有关更多信息，请参阅使用容量重新平衡来处理 Amazon EC2 竞价型实例中断 (p. 262)。

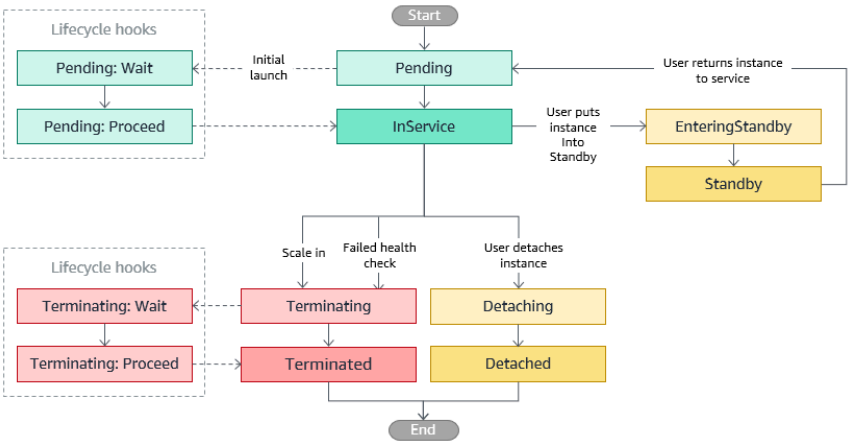
Amazon EC2 Auto Scaling 实例生命周期

Auto Scaling 组中的 EC2 实例具有的路径或生命周期不同于其他 EC2 实例中的路径或生命周期。生命周期从 Auto Scaling 组启动实例并将其投入使用时开始。生命周期在您终止实例或 Auto Scaling 组禁用实例并将其终止时结束。

Note

一旦启动实例，您就需要为实例付费，包括尚未将实例投入使用的时间。

下图阐释了 Amazon EC2 Auto Scaling 生命周期内的实例状态之间的过渡。



扩展

以下横向扩展事件指示 Auto Scaling 组启动 EC2 实例并将其附加到组：

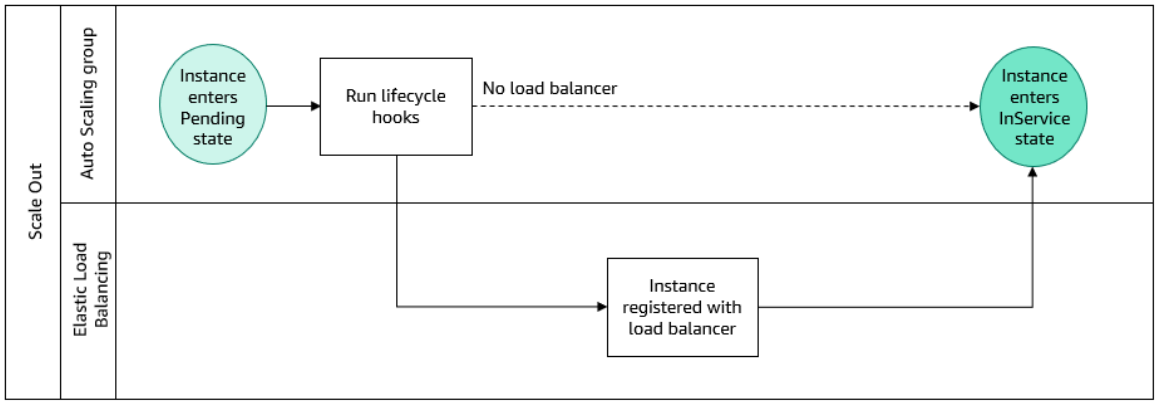
- 手动增大组的大小。有关更多信息，请参阅Amazon EC2 Auto Scaling 的手动扩缩 (p. 123)。
- 您创建一个扩展策略来自动根据指定的所需增量来增大组的大小。有关更多信息，请参阅Amazon EC2 Auto Scaling 的动态扩缩 (p. 132)。
- 您可以通过安排在某个特定时间增大组的大小来设置扩展。有关更多信息，请参阅Amazon EC2 Auto Scaling 的计划扩缩 (p. 186)。

在发生横向扩展事件时，Auto Scaling 组将使用分配的启动模板来启动所需数目的 EC2 实例。这些实例最初处于 `Pending` 状态。如果您向 Auto Scaling 组添加生命周期挂钩，则可在此处执行自定义操作。有关更多信息，请参阅生命周期钩子 (p. 9)。

在每个实例完全配置并通过 Amazon EC2 运行状况检查后，该实例将附加到 Auto Scaling 组并进入 `InService` 状态。针对 Auto Scaling 组的所需容量对实例进行计数。

如果您的 Auto Scaling 组配置为从某个 Elastic Load Balancing 负载均衡器接收流量，则 Amazon EC2 Auto Scaling 会将实例标记为 `InService`，然后自动将您的实例注册至此负载均衡器。

以下总结了为横向扩展事件向负载均衡器注册实例的工作流程。



已投入使用的实例

实例将保持 `InService` 状态，直至出现下列情况之一：

- 发生缩减事件，并且 Amazon EC2 Auto Scaling 选择终止此实例来减小 Auto Scaling 组的大小。有关更多信息，请参阅控制在横向缩减过程中要终止的 Auto Scaling 实例 (p. 222)。
- 将实例置于 `Standby` 状态。有关更多信息，请参阅进入和退出备用状态 (p. 9)。
- 您从 Auto Scaling 组分离实例。有关更多信息，请参阅分离实例 (p. 9)。
- 实例未通过所需数目的运行状况检查，因此将从 Auto Scaling 组中删除实例、终止实例和替换实例。有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。

缩小

以下缩减事件指示 Auto Scaling 组从组中分离 EC2 实例并将其终止：

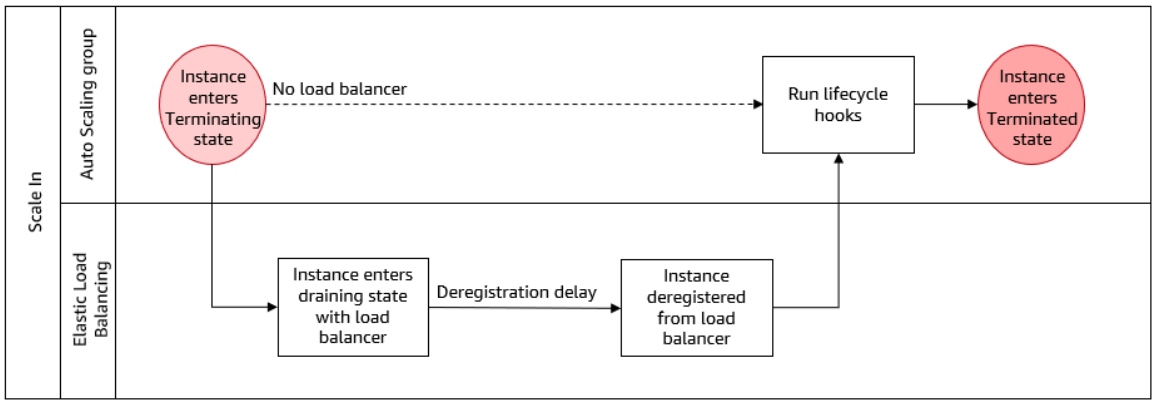
- 手动减小组的大小。有关更多信息，请参阅Amazon EC2 Auto Scaling 的手动扩缩 (p. 123)。
- 您创建一个扩展策略，自动根据指定的所需减少量来减小组的大小。有关更多信息，请参阅Amazon EC2 Auto Scaling 的动态扩缩 (p. 132)。
- 您可以通过安排在某个特定时间减小组的大小来设置扩展。有关更多信息，请参阅Amazon EC2 Auto Scaling 的计划扩缩 (p. 186)。

您必须为所创建的每个横向扩展事件创建一个相应的缩减事件。这有助于确保分配给您的应用程序的资源与对这些资源的需求尽可能相符。

发生缩减事件时，Auto Scaling 组会终止一个或多个实例。Auto Scaling 组使用其终止策略来确定要终止的实例。正在从 Auto Scaling 组中终止和关闭的实例将进入 `Terminating` 状态，且无法重新将其投入使用。如果您向 Auto Scaling 组添加生命周期挂钩，则可在此处执行自定义操作。最后，实例将完全终止并进入 `Terminated` 状态。

如果您的 Auto Scaling 组配置为从某个 Elastic Load Balancing 负载均衡器接收流量，则 Amazon EC2 Auto Scaling 会运行生命周期挂钩，然后自动从此负载均衡器取消注册终止的实例。取消注册实例可确保将所有新请求重定向到负载均衡器目标组中的其他实例，同时允许与该实例的现有连接继续，直到取消注册延迟到期。

以下总结了为横向缩减事件向负载均衡器取消注册实例的工作流程。



附加实例

您可以将符合特定条件的正在运行的 EC2 实例附加到您的 Auto Scaling 组。在附加实例后，将该实例作为 Auto Scaling 组的一部分进行管理。

有关更多信息，请参阅将 EC2 实例附上 Auto Scaling 组。 (p. 125)。

分离实例

您可以从 Auto Scaling 组中分离实例。分离实例后，您可以独立于 Auto Scaling 组管理实例或者将实例附加到其他 Auto Scaling 组。

有关更多信息，请参阅从 Auto Scaling 组中分离 EC2 实例 (p. 129)。

生命周期钩子

您可以将生命周期挂钩添加到 Auto Scaling 组，以便在实例启动或终止时执行自定义操作。

当 Amazon EC2 Auto Scaling 响应横向扩展事件时，它将启动一个或多个实例。这些实例最初处于 `Pending` 状态。如果您已将一个 `autoscaling:EC2_INSTANCE_LAUNCHING` 生命周期挂钩添加到您的 Auto Scaling 组，则实例将从 `Pending` 状态转换为 `Pending:Wait` 状态。完成生命周期操作后，实例将进入 `Pending:Proceed` 状态。在完全配置实例后，实例将附加到 Auto Scaling 组并进入 `InService` 状态。

当 Amazon EC2 Auto Scaling 响应缩减事件时，它将终止一个或多个实例。这些实例将从 Auto Scaling 组中分离并进入 `Terminating` 状态。如果您已将一个 `autoscaling:EC2_INSTANCE_TERMINATING` 生命周期挂钩添加到您的 Auto Scaling 组，则实例将从 `Terminating` 状态转换为 `Terminating:Wait` 状态。完成生命周期操作后，实例将进入 `Terminating:Proceed` 状态。在完全终止实例后，实例将进入 `Terminated` 状态。

有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

进入和退出备用状态

可以将任何处于 `InService` 状态的实例置于 `Standby` 状态。这使您能够终止对实例的使用，排查实例的问题或对实例进行更改，然后重新将实例投入使用。

处于 `Standby` 状态的实例继续由 Auto Scaling 组管理。但是，在将这些实例重新投入使用前，它们不是您的应用程序的有效部分。

有关更多信息，请参阅临时从 Auto Scaling 组中移除实例 (p. 232)。

Amazon EC2 Auto Scaling 的配额

对于每项 Amazon 服务，您的 Amazon Web Services 账户都具有默认配额（以前称为限制）。除非另有说明，否则，每个配额是区域特定的。您可以请求增加某些配额，但其他一些配额无法增加。

要查看 Amazon EC2 Auto Scaling 的配额，请打开 Service Quotas 控制台。在导航窗格中，选择 Amazon 服务，然后选择 Amazon EC2 Auto Scaling。

要请求提高配额，请参阅 Service Quotas 用户指南中的请求提高配额。如果 Service Quotas 中尚未显示配额，请使用 Auto Scaling 限制表。配额的提高会与请求所针对的区域关联。

所有申请都将提交给 Amazon Web Services Support。您可以在 Amazon Web Services Support 控制台中跟踪您的请求案例。

Amazon EC2 Auto Scaling 资源

对于您可以创建的 Auto Scaling 组和启动配置数量，您的 Amazon Web Services 账户拥有的配额如下。

资源	默认配额
每个区域的 Auto Scaling 组	500
每个区域的启动配置	200

Auto Scaling 组配置

对于 Auto Scaling 组配置，您的 Amazon Web Services 账户拥有的配额如下。无法对其进行更改。

资源	配额
每个 Auto Scaling 组的扩展策略	50
每个 Auto Scaling 组的计划操作	125
每个步进扩展策略的步进调整	20
Auto Scaling 组的生命周期挂钩	50
每个 Auto Scaling 组的 SNS 主题	10
每个 Auto Scaling 组的 Classic 负载均衡器数	50
每个 Auto Scaling 组的目标组	50

Auto Scaling 组 API 操作

Amazon EC2 Auto Scaling 提供了 API 操作，以便您通过批处理方式更改 Auto Scaling 组。有关单个操作中允许的最大项目数（最大数组成员数）的 API 限制如下。无法对其进行更改。

运算	最大数组成员数
AttachInstances	20 个实例 ID
AttachLoadBalancers	10 个负载均衡器
AttachLoadBalancerTargetGroups	10 个目标组

运算	最大数组成员数
BatchDeleteScheduledAction	50 个计划操作
BatchPutScheduledUpdateGroupAction	50 个计划操作
DetachInstances	20 个实例 ID
DetachLoadBalancers	10 个负载均衡器
DetachLoadBalancerTargetGroups	10 个目标组
EnterStandby	20 个实例 ID
ExitStandby	20 个实例 ID
SetInstanceProtection	50 个实例 ID

其他服务

其他服务（如 Amazon EC2）的配额可能会影响您的 Auto Scaling 组。有关更多信息，请参阅《亚马逊云科技一般参考》中的服务终端节点和配额。

设置 Amazon EC2 Auto Scaling

在开始使用 Amazon EC2 Auto Scaling 之前，请完成以下任务。

任务

- 准备使用 Amazon EC2 (p. 12)
- 准备使用 Amazon CLI (p. 12)

准备使用 Amazon EC2

如果您以前未使用过 Amazon EC2，请完成 Amazon EC2 文档中所述的任务。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的使用 Amazon EC2 进行设置或者适用于 Windows 实例的 Amazon EC2 用户指南中的使用 Amazon EC2 进行设置。

准备使用 Amazon CLI

可以使用 Amazon 命令行工具，在系统的命令行中发出命令来执行 Amazon EC2 Auto Scaling 和其他 Amazon 任务。

要使用 Amazon Command Line Interface（Amazon CLI），请下载、安装和配置 Amazon CLI 的版本 1 或 2。版本 1 和版本 2 中的 Amazon EC2 Auto Scaling 功能完全相同。要安装 Amazon CLI 版本 1，请参阅《Amazon CLI 用户指南》中的安装、更新和卸载 Amazon CLI。要安装 Amazon CLI 版本 2，请参阅《Amazon CLI 版本 2 用户指南》中的安装或更新 Amazon CLI 的最新版本。

Amazon CloudShell 允许您跳过在开发环境中安装 Amazon CLI 的过程，直接在 Amazon Web Services Management Console中使用它。除无需安装外，您还无需配置凭证，也不需要指定区域。您的 Amazon Web Services Management Console会话会将此上下文提供给 Amazon CLI。您可以在支持的 Amazon Web Services 区域 中使用 Amazon CloudShell。有关更多信息，请参阅使用 Amazon CloudShell 从命令行创建自动扩缩组 (p. 269)。

有关更多信息，请参阅 Amazon CLI 命令参考中的弹性伸缩。

Amazon EC2 Auto Scaling 入门

在使用 Amazon EC2 Auto Scaling 时，您必须从特定的构建块入手。本教程将引导您完成设置构建块以创建 Amazon EC2 Auto Scaling 的基本基础设施的过程。

在创建用于应用程序的自动扩缩组之前，请全面检查应用程序在 Amazon Web Services 云 中运行时的情况。请考虑以下事项：

- Auto Scaling 组应跨多少个可用区。
- 可以使用哪些现有的资源，如安全组或亚马逊系统映像 (AMI)。
- 无论您是希望进行扩展以增加或减少容量，还是只希望确保始终运行特定数量的服务器，请记住，Amazon EC2 Auto Scaling 可以同时实现这两个目标。
- 哪些指标与应用程序的性能关系最密切。
- 启动和配置服务器需要多长时间。

您越了解您的应用程序，Auto Scaling 架构的效率就越高。

> **Note**
>
> 有关简介视频，请参阅 YouTube 上的 Amazon re:Invent 2018：Amazon EC2 Auto Scaling 使容量管理变得轻松。

任务

- 演练摘要 (p. 13)
- 准备演练 (p. 14)
- 步骤 1：创建启动模板 (p. 14)
- 步骤 2：创建单个实例 Auto Scaling 组 (p. 15)
- 步骤 3：验证您的 Auto Scaling 组 (p. 16)
- 步骤 4：终止 Auto Scaling 组中的实例 (p. 16)
- 步骤 5：后续步骤 (p. 17)
- 步骤 6：清除 (p. 17)

演练摘要

在本演练中，您将：

- 创建用于定义 EC2 实例的配置模板。您可以选择启动模板或启动配置说明。虽然您可以使用启动配置，但我们建议您使用启动模板以便您可以使用 Amazon EC2 和 Amazon EC2 Auto Scaling 的最新功能。
- 在其中创建包含单个实例的 Auto Scaling 组。
- 终止实例并验证实例是否已从服务中删除并被替换。为了保持恒定数量的实例，Amazon EC2 Auto Scaling 会自动检测并响应 Amazon EC2 运行状况和可到达性检查。

如果您的 Amazon Web Services 账户是在过去 12 个月内创建的，并且还没有超出 Amazon EC2 的免费套餐权益范围，则学完本教程不需要任何费用，因为我们会帮助您选择免费套餐权益范围内的实例类型。否则，当您按照此教程操作时，您从启动实例的那一刻开始承担标准的 Amazon EC2 使用费，直至删除 Auto Scaling 组（本教程的最后一个任务），且实例状态变为 `terminated`。

准备演练

本演练假定您熟悉 EC2 实例的启动并已创建密钥对和安全组。有关更多信息，请参见适用于 Linux 实例的 Amazon EC2 用户指南中的设置 Amazon EC2。

如果您是初次使用 Amazon EC2 Auto Scaling 并希望开始使用该服务，则可以使用 Amazon Web Services 账户的默认 VPC。默认 VPC 在各个可用区中包含一个默认公有子网，以及连接到您 VPC 的 Internet 网关。您可以在 Amazon Virtual Private Cloud (Amazon VPC) 控制台的您的 VPC 页面查看您的 VPC。

步骤 1：创建启动模板

在此步骤中，使用您的 Amazon Web Services 账户凭证登录到 Amazon EC2 控制台并创建启动模板，该模板指定 Amazon EC2 Auto Scaling 为您创建的 EC2 实例的类型。包含一些信息，例如将使用的 Amazon Machine Image (AMI) 的 ID、实例类型、密钥对和安全组。

> Note
>
> 要改用启动配置，请参阅 Create a launch configuration。

创建启动模板

1. 打开 Amazon EC2 控制台的启动模板页面。
2. 在屏幕顶部的导航栏中，选择一个 Amazon Web Services 区域。您创建的启动模板和自动扩缩组将会与您所指定的区域绑定。
3. 选择Create launch template（创建启动模板）。
4. 对于 Launch template name（启动模板名称），输入 **my-template-for-auto-scaling**。
5. 在 Auto Scaling guidance（Auto Scaling 指导）下，选中复选框。
6. 对于 Application and OS Images (Amazon Machine Image) [应用程序和操作系统镜像（Amazon Machine Image）]，请从 Quick Start（快速启动）列表中选择一个 Amazon Linux 2（HVM）版本。AMI 用作实例的基本配置模板。
7. 对于 Instance type（实例类型），选择与您指定的 AMI 兼容的硬件配置。

   > Note
   >
   > 如果您的账户不到 12 个月，您可以在特定使用限制下免费使用 `t2.micro` 实例。有关更多信息，请参阅 Amazon 免费套餐。
8. （可选）对于 Key pair (login) [密钥对（登录）]，选择一个现有的密钥对。您可以使用密钥对连接包含 SSH 的 Amazon EC2 实例。有关连接到实例的内容不包括在本教程中。因此，除非您计划使用 SSH 连接到实例，否则不需要指定密钥对。
9. 对于 Network settings（网络设置）、Security groups（安全组），请选择您计划用作 Auto Scaling 组的 VPC 的同一 VPC 中的安全组。如果您没有指定安全组，实例会自动与 VPC 的默认安全组关联。
10. 您可以将 Advanced network configuration（高级网络配置）留空。如果将此设置留空，系统将创建一个主网络接口，其中具有我们为您的实例选择的 IP 地址（基于网络接口为之建立的子网）。如果您改为选择配置网络接口，则安全组必须属于该接口。
11. 选择Create launch template（创建启动模板）。
12. 在确认页面上，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

如果您当前未使用启动模板，并且不希望立即创建启动模板，则可以改为创建启动配置。

启动配置类似于启动模板，它指定 Amazon EC2 Auto Scaling 为您创建的 EC2 实例的类型。您可以包含一些信息以创建启动配置，例如，要使用的 Amazon Machine Image (AMI) 的 ID、实例类型、密钥对和安全组。

创建启动配置

1. 打开 Amazon EC2 控制台的启动配置页面。
2. 在导航栏上，选择一个 Amazon Web Services 区域。您创建的启动配置和自动扩缩组将会与您所指定的区域绑定。
3. 选择 Create launch configuration（创建启动配置），然后在 Name（名称）字段中输入 **`my-first-launch-configuration`**。
4. 对于 Amazon Machine Image (AMI)，选择 AMI。要查找特定 AMI，您可以查找合适的 AMI，记下其 ID，然后输入 ID 作为搜索条件。

   要获取 Amazon Linux 2 AMI 的 ID，请执行以下操作：

   a. 打开 Amazon EC2 控制台。
   b. 在导航窗格中的 Instances（实例）下，选择 Instances（实例），然后选择 Launch instances（启动实例）。
   c. 在 Choose an Amazon Machine Image（选择 Amazon Machine Image）页面的 Quick Start 选项卡上，请注意 Amazon Linux 2 AMI (HVM) 旁边的 AMI 的 ID。请注意，此 AMI 标记为"Free tier eligible"（符合条件的免费套餐）。
5. 对于 Instance type（实例类型），为您的实例选择硬件配置。

   Note

   如果您的账户不到 12 个月，您可以在特定使用限制下免费使用 `t2.micro` 实例。有关更多信息，请参阅 Amazon 免费套餐。
6. 在 Additional configuration（其他配置）下，对于 Advanced details（高级详细信息）、IP address type（IP 地址类型），请做出选择。要为 VPC 中的实例提供 Internet 连接，请选择一个指定了公有 IP 地址的选项。如果实例在默认 VPC 中启动，则默认分配公有 IP 地址。如果您要向实例提供 Internet 连接，但是不确定您是否有默认 VPC，请选择 Assign a public IP address to every instance（向每个实例分配公有 IP 地址）。
7. 对于安全组，请选择现有安全组。如果您保留选中创建新安全组选项，将为运行 Linux 的 Amazon EC2 实例配置默认 SSH 规则。将为运行 Windows 的 Amazon EC2 实例配置默认 RDP 角色。
8. 对于密钥对（登录），请按照指示选择密钥对选项下的选项。有关连接到实例的内容不包括在本教程中。因此，您可以选择 Proceed without a key pair（在没有密钥对的情况下继续），除非您计划使用 SSH 连接到您的实例。
9. 选择 Create launch configuration（创建启动配置）。
10. 选中新启动配置名称旁边的复选框，然后选择操作、创建 Auto Scaling 组。

步骤 2：创建单个实例 Auto Scaling 组

现在，使用 Amazon EC2 Auto Scaling 创建 Auto Scaling 组并将启动模板或启动配置添加到组。还包括实例的 VPC 子网等信息。

创建启动模板或启动配置后，使用以下过程从中断的地方继续操作。

创建 自动扩缩组

1. 在选择启动模板或配置页面上，对于 Auto Scaling 组名称，输入 **`my-first-asg`**。
2. 选择 Next（下一步）。

   此时将显示页面 Choose instance launch options（选择实例启动选项），以便选择您希望 Auto Scaling 组使用的 VPC 网络设置，并提供启动按需型实例和竞价型实例的选项（如果您选择了启动模板）。
3. 在 Network（网络）部分中，VPC 设置可保留您选择的 Amazon Web Services 区域的默认 VPC，也可选择您自己的 VPC。默认 VPC 会自动配置为向您的实例提供 Internet 连接。此 VPC 在区域的每个可用区中均包含一个公有子网。

4. 对于 Availability Zones and subnets（可用区和子网），请从您希望包含的每个可用区中选择一个子网。可以在多个可用区中使用子网以提供高可用性。有关更多信息，请参阅 选择 VPC 子网时的注意事项 (p. 302)。
5. [仅限启动模板] 在 Instance type requirements（实例类型要求）部分中，使用默认设置简化此步骤。（请勿覆盖启动模板。）在本教程中，您将仅使用启动模板中指定的实例类型启动一个按需型实例。
6. 保留本教程的其余默认值，然后选择 Skip to review（跳到审核）。

   Note

   组的初始规模由其所需容量决定。默认值为 1 实例。
7. 在 Review（审核）页面上，查看组的信息，然后选择 Create Auto Scaling group（创建 Auto Scaling 组）。

步骤 3：验证您的 Auto Scaling 组

现在，您已创建 Auto Scaling 组，可验证该组是否已启动 EC2 实例。

Tip

在以下过程中，您需要查看 Auto Scaling 组的 Activity history（活动历史记录）和 Instances（实例）部分。这两个部分都会已经显示已命名的列。要显示隐藏的列或更改显示的行数，请选择每个部分右上角的齿轮图标以打开首选项模式，根据需要更新设置，然后选择 Confirm（确认）。

验证 Auto Scaling 组已启动 EC2 实例

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中刚创建的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。可用的第一个选项卡是 Details（详细信息）选项卡，显示有关 Auto Scaling 组的信息。
3. 选择第二个选项卡，即 Activity（活动）。在 Activity history（活动历史记录）下，您可以查看与 Auto Scaling 组关联的活动的进度。Status（状态）列显示您实例的当前状态。当您的实例启动时，状态列将显示 `PreInService`。该实例启动后，状态会变为 `Successful`。您还可以使用刷新按钮来查看您的实例的当前状态。
4. 在 Instance management（实例管理）选项卡上的 Instances（实例）下，您可以查看实例的状态。
5. 验证您的实例已成功启动。启动实例只需很短的时间。

   - Lifecycle（生命周期）列显示您的实例的状态。最初，您的实例处于 `Pending` 状态。在实例准备好接收流量时，其状态为 `InService`。
   - Health status（运行状态）列显示对您的实例进行 EC2 实例运行状况检查的结果。

步骤 4：终止 Auto Scaling 组中的实例

使用这些步骤了解有关 Amazon EC2 Auto Scaling 工作原理的更多信息，特别是如何在必要时启动新实例。您在本教程中创建的 Auto Scaling 组的最小大小为一个实例。因此，如果您终止正在运行的实例，Amazon EC2 Auto Scaling 必须启动一个新实例以替换该实例。

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中您的 Auto Scaling 组旁边的复选框。
3. 在 Instance management（实例管理）选项卡上的 Instances（实例）下，选择实例的 ID。

   这时您将会跳转到 Amazon EC2 控制台中的 Instances（实例）页面，您可在其中终止实例。

4. 依次选择 Actions（操作）、Instance State（实例状态）和 Terminate（终止）。当系统提示您确认时，选择 Yes, Terminate（是，终止）。
5. 在导航窗格的 Auto Scaling 下，选择 自动扩缩组。选择您的 Auto Scaling 组，然后选择 Activity（活动）选项卡。

   Auto Scaling 组的默认冷却时间是 300 秒（5 分钟），因此您大约需要等 5 分钟才能看到扩展活动。在活动历史记录中，当扩展活动启动时，将会看到有关终止第一个实例的条目以及有关启动新实例的条目。
6. 在 Instance management（实例管理）选项卡上，Instances（实例）部分仅显示新实例。
7. 在导航窗格上的 Instances（实例）下，选择 Instances（实例）。此页面同时显示已终止的实例和新的正在运行的实例。

步骤 5：后续步骤

如果您想删除刚创建的用于自动扩展的基本基础设施，请转到下一步。否则，您可以将该基础设施作为基础，然后尝试以下一个或多个操作：

- 使用会话管理器或 SSH 连接到 Linux 实例。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用会话管理器连接到您的 Linux 实例和使用 SSH 连接到您的 Linux 实例。
- 配置 Amazon SNS 通知，以便在您的自动扩缩组启动或终止实例时收到通知。有关更多信息，请参阅使用 Amazon SNS 通知进行监控 (p. 258)。
- 手动扩展您的自动扩缩组以测试 SNS 通知。有关更多信息，请参阅手动扩展 (p. 123)。

您也可以通过阅读关于 目标跟踪扩缩策略 (p. 133) 的内容开始熟悉自动扩缩概念。如果应用程序的负载发生变化，自动扩缩组可以通过在最小和最大容量限制之间调整组的所需容量来自动横向扩展（添加实例）和横向缩减（运行更少的实例）。有关这些限制的更多信息，请参阅 为 Auto Scaling 组设置容量限制 (p. 122)。

如果您计划将负载均衡器附加到自动扩缩组，您可以学习如何使用 Amazon EC2 Auto Scaling 控制台快速创建负载均衡器。有关更多信息，请参阅从 Amazon EC2 Auto Scaling 控制台配置 Application Load Balancer 或 Network Load Balancer。 (p. 276)。

步骤 6：清除

您可以删除扩展基础设施，或者仅删除 Auto Scaling 组而保留启动模板或启动配置以供将来使用。

如果您启动的实例不在 Amazon 免费套餐范围内，则应终止实例以避免产生额外费用。当您终止实例时，与其关联的数据也将被删除。

删除 Auto Scaling 组

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中您的自动扩缩组 (`my-first-asg`) 旁边的复选框。
3. 选择 Delete (删除)。
4. 当系统提示进行确认时，键入 **`delete`** 以确认删除指定自动扩缩组，然后选择 Delete（删除）。

   名称列中的加载图标指示 Auto Scaling 组正在被删除。发生删除时，Desired（所需）、Min（最小）和 Max（最大）列显示 Auto Scaling 组具有 `0` 个实例。终止实例并删除组需要几分钟时间。刷新列表以查看当前状态。

如果要保留启动模板，请跳过以下过程。

删除启动模板

1. 打开 Amazon EC2 控制台的启动模板页面。
2. 选择您的启动模板 (`my-template-for-auto-scaling`)。
3. 选择 Actions（操作），然后选择 Delete template（删除模板）。
4. 当系统提示进行确认时，键入 **`Delete`** 以确认删除指定启动模板，然后选择 Delete（删除）。

如果您要保留启动配置，请跳过以下过程。

删除启动配置

1. 打开 Amazon EC2 控制台的启动配置页面。
2. 选择您的启动配置 (`my-first-launch-configuration`)。
3. 依次选择 Actions（操作）和 Delete launch configuration（删除启动配置）。
4. 当系统提示进行确认时，选择 Delete (删除)。

启动模板

启动模板类似于启动配置 (p. 39)，因为它指定实例配置信息。它包括 Amazon Machine Image (AMI) 的 ID、实例类型、密钥对、安全组以及用于启动 EC2 实例的其他参数。但是，定义启动模板而非启动配置可让您有多个版本的启动模板。

利用启动模板的版本控制，您可以创建全套参数的子集。然后，您可以重复使用它来创建同一启动模板的其他版本。例如，您可以创建一个启动模板，用于定义无 AMI 或用户数据脚本的基本配置。创建启动模板后，您可以创建新版本并添加具有最新版本的应用程序的 AMI 和用户数据进行测试。这将生成两个版本的启动模板。存储基本配置可帮助您保持所需的常规配置参数。您可以随时根据基本配置创建新版本的启动模板。不再需要时，您也可以删除用于测试应用程序的版本。

我们建议您使用启动模板以确保您可以访问最新功能和改进。当您使用启动配置时，并非所有的 Amazon EC2 Auto Scaling 功能都可用。例如，您无法创建 Auto Scaling 组来同时启动竞价型实例和按需型实例或者指定多个实例类型。您必须使用启动模板来配置这些功能。有关更多信息，请参阅Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。

通过启动模板，您还可以使用 Amazon EC2 的较新功能。这包括 Systems Manager 参数（AMI ID）、当前一代 EBS 预调配 IOPS 卷（io2）、EBS 卷标记、T2 无限实例、Elastic Inference 和专属主机等等。专用主机是指 EC2 实例容量供您专用的物理服务器。虽然 Amazon EC2 专用实例也在专用硬件上运行，相比专用实例，使用专用主机的优势在于，您可以自带外部供应商的合格软件许可证并在 EC2 实例上使用。

如果您当前使用启动配置，您可以通过在控制台中复制它们来将数据从现有启动配置迁移到启动模板。然后，您可以将使用启动配置的已部署 Auto Scaling 组迁移到新的启动模板。为此，请启动实例刷新以执行组的滚动更新。有关更多信息，请参阅替换 Auto Scaling 实例 (p. 94)。

> Note
>
> 有关迁移至启动模板更多信息，请参阅 Amazon 计算博客上的 Amazon EC2 Auto Scaling 将不再向启动配置添加对新 EC2 功能的支持。

创建启动模板时，所有参数都是可选的。但是，如果启动模板未指定 AMI，则无法在创建您的 Auto Scaling 组时添加 AMI。如果您指定 AMI 但没有实例类型，则可以在创建您的 Auto Scaling 组时添加一个或多个实例类型。

目录

- 权限 (p. 19)
- 为 Auto Scaling 组创建启动模板 (p. 20)
- 将启动配置复制到启动模板 (p. 26)
- 使用启动模板替换启动配置 (p. 27)
- 为容错和灵活的应用程序请求竞价型实例 (p. 28)
- 使用 Amazon Command Line Interface (Amazon CLI) 创建和管理启动模板的示例 (p. 29)
- 在启动模板中使用 Amazon Systems Manager 参数代替 AMI ID (p. 35)

权限

本节中的过程假定您已具有使用启动模板所需的权限。使用适当权限，您可以创建和管理启动模板。您也可以创建和更新 Auto Scaling 组并指定启动模板，而不是启动配置。

当您更新或创建 Auto Scaling 组并指定启动模板时，系统会检查您的 `ec2:RunInstances` 权限。如果您没有足够的权限，则会收到一条错误，指出您未获授权使用启动模板。

请求中的一些附加功能需要额外的权限，例如能够将 IAM 角色传递给预置实例或向预置实例和卷添加标签。

有关管理员如何向您授予权限的信息，请参阅 启动模板支持 (p. 328)。

为 Auto Scaling 组创建启动模板

您必须首先使用启动 EC2 实例所需的参数创建一个启动模板，然后才能使用启动模板创建 Auto Scaling 组。这些参数包括 Amazon Machine Image（AMI）的 ID 和实例类型。

启动模板提供了 Amazon EC2 Auto Scaling 的全部功能以及 Amazon EC2 的较新功能，例如当前一代 Amazon EBS 预置 IOPS 卷（io2）、EBS 卷标记、T2 无限实例、Elastic Inference 和专属主机。

要创建新的启动模板，请使用以下过程。

目录

- 创建启动模板（控制台）(p. 20)
    - 更改原定设置网络接口设置 (p. 22)
    - 修改存储配置 (p. 23)
    - 配置启动模板的高级设置 (p. 24)
- 从现有实例创建启动模板（控制台）(p. 26)
- 其他信息 (p. 26)
- 限制 (p. 26)

> **Important**
>
> 在创建启动模板时，不会完全验证启动模板参数。如果您为参数指定了错误的值，或者如果您未使用受支持的参数组合，则任何实例都无法通过此启动模板启动。确保为参数指定的值正确，并使用支持的参数组合。例如，要使用基于 Arm 的 Amazon Graviton 或 Graviton2 AMI 启动实例，您必须指定一个与 Arm 兼容的实例类型。

创建启动模板（控制台）

以下步骤说明了配置启动模板的过程：

- 指定要从其启动实例的 Amazon machine image（AMI）。
- 选择一个与您指定的 AMI 兼容的实例类型。
- 指定连接到实例时要使用的密钥对，例如使用 SSH。
- 添加一个或多个安全组以允许对实例的网络访问。
- 指定是否要为每个实例挂载附加卷。
- 将自定义标签（键值对）添加到实例和卷。

创建启动模板

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的实例下，选择启动模板。
3. 选择 Create launch template (创建启动模板)。为启动模板的初始版本输入名称并提供描述。

4. 在 Auto Scaling 指导下，选中复选框让 Amazon EC2 提供指导，以帮助创建要与 Amazon EC2 Auto Scaling 结合使用的模板。
5. 在 Launch template contents（启动模板内容）下，填写每个必填字段以及所有可选字段。

   a. Application and OS Images (Amazon Machine Image) [应用程序和操作系统镜像（Amazon Machine Image）]：（必填项）选择实例 AMI 的 ID。您可以搜索所有可用的 AMI，也可以从 Recents（最近）或 Quick Start（快速启动）列表中选择一个 AMI。如果您没有看到所需的 AMI，请选择 Browser more AMIs（浏览更多 AMI）以浏览完整的 AMI 目录。

      要选择自定义 AMI，您必须首先从某个自定义实例创建该 AMI。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的创建 AMI

   b. 对于 Instance type（实例类型），请选择与您指定的 AMI 兼容的单个实例类型。

      如要启动包括多种实例类型的 Auto Scaling 组，请选择 Advanced（高级）、Specify instance type attributes（指定实例类型属性），然后指定以下选项：

      - Number of vCPUs（vCPU 数）：输入最小和最大 vCPU 数。若要表示为无限制，请输入最小值为 0，然后将最大值留空。
      - Amount of memory (MiB) [内存大小（MiB）]：输入最小和最大内存大小（以 MiB 为单位）。若要表示为无限制，请输入最小值为 0，然后将最大值留空。
      - 展开 Optional instance type attributes（可选的实例类型属性），然后选择 Add attribute（添加属性）以进一步限制可用于满足所需容量的实例类型。有关每个属性的信息，请参阅 Amazon EC2 API 参考中的 InstanceRequirementsRequest。
      - Resulting instance types（结果实例类型）：您可以查看符合指定计算要求的实例类型，例如 vCPU、内存和存储。
      - 要排除实例类型，请选择 Add Attribute（添加属性）。从 Attribute（属性）列表中，选择 Excluded instance types（排除的实例类型）。从 Attribute value（属性值）列表中，选择要排除的实例类型。

      有关更多信息，请参阅示例：使用基于属性的实例类型选择创建 Auto Scaling 组 (p. 72)。

   c. Key pair (login) [密钥对（登录）]：对于 Key pair name（密钥对名称），请选择一个现有密钥对，或选择 Create new key pair（创建新密钥对）以新建一个密钥对。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 密钥对和 Linux 实例。

   d. Network settings（网络设置）：对于 Firewall (security groups) [防火墙（安全组）]，请使用一个或多个安全组，或将此留空并将一个或多个安全组配置为网络接口的一部分。有关更多信息，请参阅《适用于 Linux 实例Amazon EC2 用户指南》 中的适用于 Linux 实例的 Amazon EC2 安全组。

      如果您未在启动模板中指定任何安全组，则 Amazon EC2 将使用您的 Auto Scaling 组将在其中启动实例的 VPC 的原定设置安全组。预设情况下，此安全组不允许来自外部网络的入站流量。有关更多信息，请参阅《Amazon VPC 用户指南》中的您的 VPC 的原定设置安全组。

   e. 请执行下列操作之一：

      - 更改原定设置网络接口设置。例如，您可以启用或禁用公有 IPv4 寻址功能，该功能会覆盖子网上的自动分配公有 IPv4 地址设置。有关更多信息，请参阅更改原定设置网络接口设置 (p. 22)。
      - 跳过此步骤以保留原定设置网络接口设置。

   f. 请执行下列操作之一：

      - 修改存储配置。有关更多信息，请参阅修改存储配置 (p. 23)。
      - 跳过此步骤以保留原定设置存储配置。

   g. 对于 Resource tags（资源标签），请提供键值组合以指定标签。如果您在启动模板中指定了实例标签，然后选择将 Auto Scaling 组的标签传播到其实例，则所有标签都会合并。如果为启动模板中的标签和 Auto Scaling 组中的标签指定了相同的标签键，则优先使用该组中的标签值。

6. （可选）配置高级设置。有关更多信息，请参阅配置启动模板的高级设置 (p. 24)。
7. 准备好创建启动模板后，请选择 Create launch template（创建启动模板）。

8. 要创建 Auto Scaling 组，请从确认页面上选择创建 Auto Scaling 组。

更改原定设置网络接口设置

这一部分说明了如何更改原定设置网络接口设置。例如，您可以定义是否要为每个实例分配一个公有 IPv4 地址，而不是默认为子网上的自动分配公有 IPv4 地址设置。

注意事项和限制

更改原定设置网络接口设置时，请记住以下注意事项和限制：

- 您必须将安全组配置为该网络接口的一部分，而不是在模板的 Security groups（安全组）部分中配置。您不能在这两处指定安全组。
- 您不能为网络接口分配其他私有 IP 地址，即所谓的 secondary IP addresses（辅助私有 IP 地址）。
- 如果指定现有的网络接口 ID，则只能启动一个实例。为此，必须使用 Amazon CLI 或 SDK 创建 Auto Scaling 组。创建组时，必须指定可用区，但不指定子网 ID。此外，仅当现有网络接口的设备索引为 0 时才可指定该接口。
- 如果您指定多个网络接口，则不能自动分配公有 IPv4 地址。也无法跨网络接口指定重复的设备索引。主网络接口和辅助网络接口都驻留在同一子网中。有关更多信息，请参阅使用 Amazon VPC 为 Auto Scaling 实例提供网络连接 (p. 300)。
- 实例启动时，系统会自动为每个网络接口分配一个私有地址。该地址来自启动实例的子网的 CIDR 范围。有关为 VPC 或子网指定 CIDR 范围（或 IP 地址范围）的信息，请参阅 Amazon VPC 用户指南。

更改原定设置网络接口设置

1. 在 Network settings（网络设置）下，展开 Advanced network configuration（高级网络配置）。
2. 选择 Add network interface（添加网络接口）以配置主网络接口，同时应注意以下字段：

   a. Device index（设备索引）：保留原定设置值 0 以将您的更改应用于主网络接口（eth0）。

   b. Network interface（网络接口）：保留原定设置值 New interface（新接口），以让 Amazon EC2 Auto Scaling 在启动实例时自动创建一个新的网络接口。您也可以选择一个设备索引为 0 的现有可用网络接口，但这将您的 Auto Scaling 组限定为单个实例。

   c. Description（描述）：（可选）请输入一个描述性的名称。

   d. Subnet（子网）：保留原定设置 Don't include in launch template（不包括在启动模板中）。

      如果 AMI 指定了子网的网络接口，则会导致错误。我们建议关闭 Auto Scaling guidance（Auto Scaling 指南）以解决此问题。进行此更改后，您将不会收到错误消息。但无论指定的子网来自何处，Auto Scaling 组的子网设置将会优先，且不能被覆盖。

   e. Auto-assign public IP（自动分配公有 IP）：更改设备索引为 0 的网络接口是否会收到公有 IPv4 地址。预设情况下，原定设置子网中的实例会收到一个公有 IPv4 地址，而非原定设置子网中的实例不会收到。选择启用或禁用可以覆盖子网的默认设置。

   f. Security groups（安全组）：为网络接口选择一个或多个安全组。对于 Auto Scaling 组会将实例启动到其中的 VPC，必须为其配置各个安全组。有关更多信息，请参阅《适用于 Linux 实例Amazon EC2 用户指南》 中的适用于 Linux 实例的 Amazon EC2 安全组。

   g. Delete on termination（终止时删除）：选择 Yes（是）以在实例终止时删除网络接口，或选择 No（否）以保留网络接口。

   h. Elastic Fabric Adapter：要支持高性能计算（HPC）使用案例，请将网络接口更改为某个 Elastic Fabric Adapter 网络接口。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 Elastic Fabric Adapter。

   i. Network card index（网卡索引）：选择 0 以将主要网络接口挂载到设备索引为 0 的网卡。如果此选项不可用，则保留原定设置 Don't include in launch template（不包括在启动模板中）。将网络接口挂载到特定网卡的功能仅适用于支持的实例类型。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的网卡。

3. 要添加辅助网络接口，请选择添加网络接口。

修改存储配置

您可以修改从由 Amazon EBS-backed AMI 或实例存储支持的 AMI 启动的实例的存储配置。您还可以指定要挂载到实例的附加 EBS 卷。AMI 会包含一个或多个存储卷，包括根卷 [Volume 1 (AMI Root)] [卷 1（AMI 根）]。

修改存储配置

1. 在 Configure storage（配置存储）中，修改卷的大小或类型。

   如果指定的卷大小值超出相应卷类型的限制，或者小于快照大小，则会显示错误消息。此消息会提供有关该字段可以接受的最小值或最大值信息，以帮助您解决此问题。

   这时仅会显示与由 Amazon EBS-backed AMI 关联的卷。要显示从实例存储支持的 AMI 启动的实例的存储配置信息，请选择 Instance store volumes（实例存储卷）部分的 Show details（显示详细信息）。

   要指定所有 EBS 卷参数，请切换到右上角的 Advanced（高级）视图。
2. 对于高级选项，请展开要修改的卷并按如下方式配置该卷：

   a. Storage type（存储类型）：要与实例关联的卷类型（EBS 或临时卷）。仅当您选择支持实例存储（临时）卷类型的实例类型时，该卷类型才可用。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 Amazon EC2 实例存储和 Amazon EBS 卷。

   b. Device name（设备名称）：从卷的可用设备名称列表中进行选择。

   c. Snapshot（快照）：选择要从其中创建卷的快照。您可以通过在 Snapshot（快照）字段中输入文本来搜索可用的共享快照和公有快照。

   d. Size (GiB)（大小 (GiB)）：对于 EBS 卷，您可以指定存储大小。如果您选择了有资格享用免费套餐的 AMI 和实例，请记住，若要享用免费套餐，您必须将总存储大小保持为 30GiB 以下。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的针对 EBS 卷的大小和配置的限制。

   e. Volume type（卷类型）：对于 EBS 卷，请选择该卷类型。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 Amazon EBS 卷类型。

   f. IOPS：如果您已选择预置 IOPS SSD（io1 和 io2）以及通用型 SSD（gp3）卷类型，则您可以输入卷可支持的每秒输入/输出操作数（IOPS）。这对于 io1、io2 和 gp3 卷是必需的。gp2、st1、sc1 或标准卷不支持。

   g. Delete on termination（终止时删除）：对于 EBS 卷，选择 Yes（是）以在终止实例时删除此卷或选择 No（否）以保留此卷。

   h. Encrypted（加密）：如果实例类型支持 EBS 加密，则可以选择 Yes（是）以为此卷启用加密。如果默认情况下在此区域中启用了加密，则会为您启用加密。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EBS 加密和原定设置加密。

      设置此参数的默认效果会随所选卷源而异，如下表所述。在所有情况下，您都必须拥有使用指定 Amazon KMS key 密钥的权限。

      加密结果

如果 **Encrypted** 参数设置为...	如果卷源...	则默认加密状态为...	注意
否	新（空）卷	未加密*	不适用
	您拥有的未加密快照	未加密*	

如果 **Encrypted** 参数设置为...	如果卷源...	则默认加密状态为...	注意
	您拥有的加密快照	按相同密钥加密	
	与您共享的未加密快照	未加密*	
	与您共享的加密快照	由默认 KMS 密钥加密	
是	新卷	由默认 KMS 密钥加密	要使用非原定设置 CMK 密钥，请为 KMS key（KMS 密钥）参数指定一个值。
	您拥有的未加密快照	由默认 KMS 密钥加密	
	您拥有的加密快照	按相同密钥加密	
	与您共享的未加密快照	由默认 KMS 密钥加密	
	与您共享的加密快照	由默认 KMS 密钥加密	

* 如果启用了原定设置加密，则所有新创建的卷 [不论是否将 Encrypted（加密）参数设置为 Yes（是）] 都将使用原定设置 KMS 密钥。如果您同时设置了 Encrypted（加密）和 KMS key（KMS 密钥）参数，则可指定非原定设置 KMS 密钥。

i. KMS key（KMS 密钥）：如果您为 Encrypted（加密）选择的是 Yes（是），则必须选择一个客户管理的密钥来加密该卷。如果默认情况下在此区域中启用了加密，则将为您选择默认的客户托管密钥。您可以选择其他密钥，或指定您之前使用 Amazon Key Management Service 创建的任何客户管理的密钥的 ARN。

3. 要指定附加卷以挂载到此启动模板启动的实例，请选择 Add new volume（添加新卷）。

配置启动模板的高级设置

您可以定义 Auto Scaling 实例所需的任何其他功能。例如，您可以选择一个 IAM 角色，以供您的应用程序在访问其他 Amazon 资源或指定实例启动后可用于执行常见自动配置任务的实例用户数据。

以下步骤讨论了需要注意的最有用的设置。有关任何设置的更多信息，请参阅高级详细信息下适用于 Linux 实例的 Amazon EC2 用户指南中的创建启动模板。

配置高级设置

1. 对于高级详细信息，展开该节以查看字段。
2. 对于购买选项，您可以选择选择请求竞价型实例可按照 Spot 价格请求 Spot 实例，以按需价格为上限；而选择自定义可更改默认竞价型实例设置。对于 Auto Scaling 组，必须指定不带结束日期的一次性请求（原定设置）。有关更多信息，请参阅为容错和灵活的应用程序请求竞价型实例 (p. 28)。

   Note

   Amazon EC2 Auto Scaling 使您能够覆盖启动模板中的实例类型，以创建使用多种实例类型并启动竞价型实例和按需型实例的 Auto Scaling 组。为此，您必须使启动模板中的 Purchasing option（购买选项）保持未指定状态。
   如果您尝试在已指定 Purchasing option（购买选项）的情况下使用启动模板创建混合实例组，您将收到以下错误。

   ```
   Incompatible launch template: You cannot use a launch template that
   is set to request Spot Instances (InstanceMarketOptions) when you
   configure an Auto Scaling group with a mixed instances policy. Add a
   different launch template to the group and try again.
   ```

   有关创建混合实例组的信息，请参阅 Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。

3. 对于 IAM 实例配置文件，您可以指定 Amazon Identity and Access Management (IAM) 实例配置文件以与实例关联。当您选择实例配置文件时，会将对应的 IAM 角色与 EC2 实例关联。有关更多信息，请参阅 在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)。
4. 对于终止保护，选择是否防止实例意外终止。当您启用终止保护时，它会提供额外的终止保护，但不会防止 Amazon EC2 Auto Scaling 启动终止。要控制 Auto Scaling 组是否可以终止特定实例，请使用 实例横向缩减保护 (p. 230)。
5. 对于详细的 CloudWatch 监控，选择是否启用实例，以 1 分钟的间隔向 Amazon CloudWatch 发布指标数据。将收取额外费用。有关更多信息，请参阅配置 Auto Scaling 实例的监控 (p. 253)。
6. 对于 Elastic inference（弹性推理）：请选择要附加到 EC2 CPU 实例的弹性推理加速器。将收取额外费用。有关更多信息，请参阅 Amazon Elastic Inference 开发人员指南 中的使用 Amazon Elastic Inference。
7. 对于 T2/T3 无限，选择是否允许应用程序突增到基准以上所需的时间。此字段仅适用于 T2、T3 和 T3a 实例。可能收取额外费用。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的使用 Auto Scaling 组以“无限”模式启动可突增性能实例。
8. 对于置放群组名称，指定在其中启动实例的置放群组。并非可以在置放群组中启动所有实例类型。如果使用某个指定其他置放群组的 CLI 命令配置 Auto Scaling 组，则 Auto Scaling 组的置放群组将会优先。
9. 对于容量预留，您可以指定是在共享容量中、任何 open 容量预留中、特定容量预留中还是容量预留组中启动实例。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的在现有容量预留中启动实例。
10. 对于租赁，您可以选择在共享硬件（共享）、专用硬件（专用）上运行您的实例，或者当使用主机资源组时，也可以在专用主机（专用主机）上运行您的实例。可能收取额外费用。

    如果您选择了专用主机，请填写以下信息：

    - 对于租赁主机资源组，您可以为 BYOL AMI 指定要在专用主机上使用的主机资源组。在使用此功能之前，您不必先在您的账户中分配专用主机。无论何种情况，您的实例都会在专用主机上自动启动。请注意，基于许可证配置关联的 AMI 一次只能映射到一个主机资源组。有关更多信息，请参阅 Amazon License Manager 用户指南中的主机资源组。
11. 对于许可证配置，您可以指定要使用的许可证配置。您可以根据指定的许可证配置启动实例，以跟踪您的许可证使用情况。有关更多信息，请参阅 License Manager 用户指南中的创建许可证配置。
12. 要为与此版本的启动模板关联的所有实例配置实例元数据选项，请执行以下操作：

    a. 对于元数据可访问，选择是启用还是禁用对实例元数据服务的 HTTP 终端节点的访问。预设情况下，将启用 HTTP 终端节点。如果您选择禁用终端节点，则会关闭对实例元数据的访问。只有在启用 HTTP 终端节点时，才可以指定要求 IMDSv2 的条件。

    b. 对于元数据版本，您可以选择在请求实例元数据时要求使用实例元数据服务版本 2 (IMDSv2)。如果您不指定值，原定设置为同时支持 IMDSv1 和 IMDSv2。

    c. 对于元数据标记响应跃点限制，您可以为元数据标记设置允许的网络跃点数。如果您未指定值，则原定设置为 1。

    有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的配置实例元数据服务。
13. 对于 User data（用户数据），您可以添加 shell 脚本和 cloud-init 指令，以在启动时自定义实例。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 启动时对 Linux 实例运行命令。

    Note

    在启动时运行脚本会增加使实例做好准备以供使用所需的时间量。但是，您可以通过向 Auto Scaling 组添加生命周期钩子，在实例进入 `InService` 状态之前为脚本完成留出额外的时间。有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。
14. 选择创建启动模板。
15. 要创建 Auto Scaling 组，请从确认页面上选择创建 Auto Scaling 组。

从现有实例创建启动模板（控制台）

通过现有实例创建启动模板

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格上的 Instances（实例）下，选择 Instances（实例）。
3. 选择实例，然后依次选择操作、图像和模板、从实例创建模板。
4. 提供名称和说明。
5. 在 Auto Scaling 指导下，选中复选框。
6. 根据需要调整任何参数，然后选择创建启动模板。
7. 要创建 Auto Scaling 组，请从确认页面上选择创建 Auto Scaling 组。

其他信息

有关创建启动模板的附加信息，请参阅：

- 适用于 Linux 实例的 Amazon EC2 用户指南的从启动模板启动实例部分
- 《Amazon CloudFormation 用户指南》的 Auto Scaling 模板代码段部分
- Amazon CloudFormation 用户指南的 AWS::EC2::LaunchTemplate 部分

有关创建具有启动模板的 Auto Scaling 组的过程，请参阅以下主题：

- 使用启动模板创建 Auto Scaling 组 (p. 78)
- Auto Scaling 组具有多个实例类型和购买选项 (p. 50)
- 示例：使用基于属性的实例类型选择创建 Auto Scaling 组 (p. 72)

限制

- Amazon EC2 允许您在启动模板中配置子网。然而，该 Auto Scaling 组的子网设置将会优先于启动模板的子网设置。
- 由于将忽略启动模板中的子网设置以支持在 Auto Scaling 组中指定的设置，因此为给定实例创建的所有网络接口都将连接到与实例相同的子网。有关用户定义网络接口的其他限制，请参阅 更改原定设置网络接口设置 (p. 22)。
- 启动模板允许您在 Auto Scaling 组中配置其他设置以启动多个实例类型并组合按需型和竞价型购买选项，如 Auto Scaling 组具有多个实例类型和购买选项 (p. 50) 中所述。如果在启动模板中指定竞价型实例请求，则不支持具有这种组合的实例启动。
- 只有在指定主机资源组时，才可支持专用主机（主机租赁）。您不能定位特定主机 ID 或使用主机放置关联。

将启动配置复制到启动模板

要从启动配置迁移到启动模板，必须将启动配置复制或重新创建为启动模板。我们建议您迁移到启动模板以利用 Amazon EC2 和 Amazon EC2 Auto Scaling 的最新功能。

如果复制启动配置，您可以一次性迁移所有这些配置，也可以通过选择要复制的启动配置来执行随时间推移的增量迁移。复制功能只能从控制台提供。

将启动配置复制到启动模板（控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的 Auto Scaling (自动调整) 上，选择 Launch Configurations (启动配置)。
3. 选择您要复制的启动配置，然后选择复制到启动模板，复制所选。这将设置一个与您所选启动配置具有相同名称和选项的新启动模板。
4. 对于新建启动模板名称，您可以使用启动配置的名称（默认值）或者输入新名称。启动模板名称必须是唯一的。
5. （可选）若要使用新启动模板创建 Auto Scaling 组，请选择使用新模板创建 Auto Scaling 组。
6. 选择 Copy（复制）。

如果您知道要将所有启动配置复制到启动模板，请按以下步骤操作。

将所有启动配置复制到启动模板（控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的 Auto Scaling (自动调整) 上，选择 Launch Configurations (启动配置)。
3. 选择复制到启动模板，全部复制。这将当前地区中的每个启动配置复制到具有相同名称和选项的新启动模板。
4. 选择 Copy（复制）。

接下来，您可以更新现有 Auto Scaling 组以指定您创建的启动模板。有关更多信息，请参阅 使用启动模板替换启动配置 (p. 27)。作为另一个选项，您可以按照 基于实例刷新替换 Auto Scaling 实例 (p. 95) 中的过程将新启动模板添加到 Auto Scaling 组并立即更新您的 Auto Scaling 实例。

使用启动模板替换启动配置

编辑具有现有启动配置的 Auto Scaling 组时，您可以选择使用启动模板替换启动配置。这使您可以将启动模板用于当前使用的任意 Auto Scaling 组。在这样做的过程中，您可以利用启动模板的版本控制和其他功能。

替换 Auto Scaling 组的启动配置后，使用新的启动模板启动任何新实例。现有实例不会受到影响。要更新现有实例，请终止这些实例，以便 Auto Scaling 组替换这些实例，或者根据您的终止策略 (p. 222)启用自动扩展，以逐步使用较新实例替换较早实例。

> **Note**
>
> 借助实例刷新功能，您可以替换 Auto Scaling 组中的实例，以立即启动使用启动模板的新实例。有关更多信息，请参阅 基于实例刷新替换 Auto Scaling 实例 (p. 95)。

先决条件

您必须先创建启动模板，然后才能替换 Auto Scaling 组中的启动配置。创建启动模板的基本方法是从启动配置进行复制。有关更多信息，请参阅将启动配置复制到启动模板 (p. 26)。

如果您将 Auto Scaling 组切换为使用启动配置，请确保您的权限是最新的。为了使用启动模板，您需要特定的权限。

替换 Auto Scaling 组的启动配置（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在页面底部打开一个拆分窗格，其中显示有关所选组的信息。
3. 在详细信息选项卡上，选择启动配置、编辑。

4. 选择切换到启动模板。
5. 对于启动模板，选择您的启动模板。
6. 对于版本，根据需要选择启动模板版本。在创建启动模板版本之后，您可以选择 Auto Scaling 组在扩展时是使用启动模板的默认版本还是最新版本。
7. 完成后，选择更新。

使用命令行替换启动配置

您可以使用以下任一命令：

- update-auto-scaling-group (Amazon CLI)
- Update-ASAutoScalingGroup (Amazon Tools for Windows PowerShell)

有关使用 CLI 命令更新 Auto Scaling 组以使用启动模板的示例，请参阅 更新 Auto Scaling 组以使用启动模板 (p. 34)。

为容错和灵活的应用程序请求竞价型实例

在启动模板中，您可以选择请求没有结束日期或持续时间的竞价型实例。与 EC2 按需价格相比，Amazon EC2 竞价型实例是以极大折扣提供的备用容量。如果能灵活控制应用程序的运行时间并且应用程序可以中断，竞价型实例就是经济实惠之选。有关创建请求竞价型实例的启动模板的更多信息，请参阅 配置启动模板的高级设置 (p. 24)。

> **Important**
>
> 竞价型实例通常用于补充按需实例。对于此情景，您可以将用于启动竞价型实例的相同设置指定为 Auto Scaling 组的设置的一部分。当您将设置指定为 Auto Scaling 组的一部分时，您只能在启动一定数量的按需实例后请求启动竞价型实例，然后在组扩展时继续启动按需实例和竞价型实例的某些组合。有关更多信息，请参阅 Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。

本主题介绍如何通过在启动模板中指定设置而不是在 Auto Scaling 组中指定设置，仅在 Auto Scaling 组中启动竞价型实例。本主题中的信息也适用于请求带启动模板 (p. 39)的竞价型实例的 Auto Scaling 组。不同之处在于启动配置需要最高价，但对于启动模板，最高价是可选的。

在创建启动模板以仅启动竞价型实例时，请注意以下事项：

- Spot 价格。您只需为您启动的竞价型实例支付当前 Spot 价格。此定价会根据长期供需趋势缓慢发生变化。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的竞价型实例和定价和节省成本。
- 设置您的最高价。您可以选择在启动模板中包含竞价型实例的每小时最高价。如果具有可用的容量，并且您的最高价超过当前 Spot 价格，Amazon EC2 竞价服务将立即完成您的请求。如果竞价型实例的价格超过 Auto Scaling 组中正在运行的实例的最高价，它会终止实例。

  > **Warning**
  >
  > 如果您未收到任何竞价型实例 (例如当您的最高价太低时)，您的应用程序可能不运行。要尽可能长时间利用可用的竞价型实例，请将最高价设置为接近按需价格。
- 在可用区之间平衡。如果您指定多个可用区，Amazon EC2 Auto Scaling 会在指定可用区之间分配竞价请求。如果您的最高价在一个可用区中过低而无法执行任何请求，Amazon EC2 Auto Scaling 会检查请求是否已在其他区域中被执行。如果是，Amazon EC2 Auto Scaling 会取消失败的请求并在已执行请求的可用区之间重新分配它们。如果没有执行请求的可用区中的价格降得足够低以使未来的请求能够成功，Amazon EC2 Auto Scaling 会在所有可用区之间重新平衡。
- 竞价型实例终止。竞价型实例可以随时终止。随着竞价型实例可用性或价格的变化，Amazon EC2 Spot 服务可以终止 Auto Scaling 组中的竞价型实例。扩展或执行运行状况检查时，Amazon EC2 Auto Scaling 能够通过终止按需实例的相同方式终止竞价型实例。当实例终止时，任何存储都将被删除。

- 保持所需容量。竞价型实例终止后，Amazon EC2 Auto Scaling 会尝试启动另一个竞价型实例来保持该组的所需容量。如果最高价高于当前 Spot 价格，则会启动竞价型实例。如果竞价型实例请求失败，它将继续尝试。
- 更改您的最高价。要更改最高价，请创建新的启动模板或使用新的最高价更新现有启动模板，然后将其与 Auto Scaling 组关联。只要现有竞价型实例所用的启动模板中指定的最高价高于当前 Spot 价格，这些实例就会继续运行。如果没有设置最高价，则默认最高价为按需价格。

使用 Amazon Command Line Interface (Amazon CLI) 创建和管理启动模板的示例

您可以通过 Amazon Web Services Management Console、Amazon CLI 或 SDK 创建并管理启动模板。本节显示了从 Amazon CLI 为 Amazon EC2 Auto Scaling 创建和管理启动模板的实例。

目录

- 示例用法 (p. 29)
- 创建基本的启动模板 (p. 30)
- 指定在启动时标记实例的标签 (p. 30)
- 指定要传递到实例的 IAM 角色 (p. 30)
- 分配公有 IP 地址 (p. 31)
- 指定用于在启动时配置实例的用户数据脚本 (p. 31)
- 指定块储存设备映射 (p. 31)
- 指定专属主机以从外部供应商获得软件许可证 (p. 31)
- 指定现有网络接口 (p. 31)
- 创建多个网络接口 (p. 32)
- 管理启动模板 (p. 32)
- 更新 Auto Scaling 组以使用启动模板 (p. 34)

示例用法

```
{
    "LaunchTemplateName": "my-template-for-auto-scaling",
    "VersionDescription": "test description",
    "LaunchTemplateData": {
        "ImageId": "ami-04d5cc9b88example",
        "InstanceType": "t2.micro",
        "SecurityGroupIds": [
            "sg-903004f88example"
        ],
        "KeyName": "MyKeyPair",
        "Monitoring": {
            "Enabled": true
        },
        "Placement": {
            "Tenancy": "dedicated"
        },
        "CreditSpecification": {
            "CpuCredits": "unlimited"
        },
        "MetadataOptions": {
            "HttpTokens": "required",
            "HttpPutResponseHopLimit": 1,
            "HttpEndpoint": "enabled"
```

```
        }
    }
}
```

创建基本的启动模板

要创建基本的启动模板，请使用以下 create-launch-template 命令，并进行这些修改：

- 将 ami-04d5cc9b88example 替换为要从其中启动实例的 AMI 的 ID。
- 将 t2.micro 替换为与指定的 AMI 兼容的实例类型。

此示例创建了名为 *my-template-for-auto-scaling* 的启动模板。如果在默认 VPC 中启动由此启动模板创建的实例，则默认情况下它们会收到一个公有 IP 地址。如果在非默认 VPC 中启动实例，则默认情况下它们不会收到一个公有 IP 地址。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

有关引用 JSON 格式参数的更多信息，请参阅 Amazon Command Line Interface 用户指南中的在 Amazon CLI 中将引号和字符串结合使用。

或者，您可以在配置文件中指定 JSON 格式参数。

以下示例创建了基本的启动模板，其中引用了启动模板参数值的配置文件。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data file://config.json
```

config.json 的内容：

```
{
    "ImageId":"ami-04d5cc9b88example",
    "InstanceType":"t2.micro"
}
```

指定在启动时标记实例的标签

以下示例在启动时将标签（例如，purpose=webserver）添加到实例。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"TagSpecifications":[{"ResourceType":"instance","Tags":
[{"Key":"purpose","Value":"webserver"}]}],"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

> **Note**
>
> 如果您在启动模板中指定了实例标签，然后选择将 Auto Scaling 组的标签传播到其实例，则所有标签都会合并。如果为启动模板中的标签和 Auto Scaling 组中的标签指定了相同的标签键，则优先使用该组中的标签值。

指定要传递到实例的 IAM 角色

以下示例指定启动时要传递给实例的与 IAM 角色关联的实例配置文件的名称。有关更多信息，请参阅在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
--launch-template-data '{"IamInstanceProfile":{"Name":"my-instance-
profile"},"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

分配公有 IP 地址

以下 create-launch-template 示例配置了启动模板，将公有地址分配给在非默认 VPC 中启动的实例。

Note

指定网络接口时，请为 Groups 指定值（对应于 Auto Scaling 组将实例启动到其中的 VPC 的安全组）。指定 VPC 子网作为 Auto Scaling 组的属性。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":
[{"DeviceIndex":0,"AssociatePublicIpAddress":true,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true}],"ImageId":"ami-04d5cc9b88example","InstanceType":'
```

指定用于在启动时配置实例的用户数据脚本

以下示例将用户数据脚本指定为 base64 编码的字符串，用于在启动时配置实例。create-launch-template 命令需要 base64 编码的用户数据。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
--launch-template-data
 '{"UserData":"IyEvYmluL2Jhc...","ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

指定块储存设备映射

以下 create-launch-template 示例创建了具有块设备映射的启动模板：映射到 /dev/xvdcz 的 22-gigabyte EBS 卷。/dev/xvdcz 卷使用通用型 SSD (gp2) 卷类型，在它连接的实例终止时会被删除。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"BlockDeviceMappings":[{"DeviceName":"/dev/xvdcz","Ebs":
{"VolumeSize":22,"VolumeType":"gp2","DeleteOnTermination":true}}],"ImageId":"ami-04d5cc9b88example","Ir
```

指定专属主机以从外部供应商获得软件许可证

如果您指定主机租赁，您可以指定主机资源组和 License Manager 许可证配置，以便从外部供应商获得合格的软件许可证。然后，您可以通过使用以下 create-launch-template 命令来在 EC2 实例上使用许可证。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"Placement":
{"Tenancy":"host","HostResourceGroupArn":"arn"},"LicenseSpecifications":
[{"LicenseConfigurationArn":"arn"}],"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

指定现有网络接口

以下 create-launch-template 示例将主网络接口配置为使用现有网络接口。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":[{"DeviceIndex":0,"NetworkInterfaceId":"eni-
b9a5ac93","DeleteOnTermination":false}],"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

创建多个网络接口

以下 create-launch-template 示例添加了辅助网络接口。主网络接口的设备索引为 0，而辅助网络接口的设备索引为 1。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":[{"DeviceIndex":0,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true},{"DeviceIndex":1,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true}],"ImageId":"ami-04d5cc9b88example","InstanceType":'
```

如果您使用支持多个网卡和 Elastic Fabric Adapters (EFA) 的实例类型，则可以向辅助网卡添加辅助接口并使用以下 create-launch-template 命令来启用 EFA。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的将 EFA 添加到启动模板。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":
[{"NetworkCardIndex":0,"DeviceIndex":0,"Groups":
["sg-7c2270198example"],"InterfaceType":"efa","DeleteOnTermination":true},
{"NetworkCardIndex":1,"DeviceIndex":1,"Groups":
["sg-7c2270198example"],"InterfaceType":"efa","DeleteOnTermination":true}],"ImageId":"ami-09d95fab7fexa
```

> **Warning**
>
> p4d.24xlarge 实例类型产生的成本高于本节中的其他示例。有关 P4d 实例定价的更多信息，请参阅 Amazon EC2 P4d 实例定价。

> **Note**
>
> 将来自同一子网的多个网络接口附加到一个实例可能会引入非对称路由，尤其是在使用非 Amazon Linux 变体的实例上。如果需要此类配置，则必须在操作系统中配置辅助网络接口。有关示例，请参阅 Amazon 知识中心中的如何在我的 Ubuntu EC2 实例中使用我的辅助网络接口？。

管理启动模板

Amazon CLI 包括其他几个可帮助您管理启动模板的命令。

目录

- 列出并描述启动模板 (p. 32)
- 创建启动模板版本 (p. 34)
- 删除启动模板版本 (p. 34)
- 删除启动模板 (p. 34)

列出并描述启动模板

您可以使用两个 Amazon CLI 命令获取有关启动模板的信息：describe-launch-templates 和 describe-launch-template-versions。

describe-launch-templates 命令可让您获取已创建的任何启动模板的列表。您可以使用选项筛选启动模板名称、创建时间、标签键或标记键值组合的结果。此命令可返回有关任何启动模板的汇总信息，包括启动模板标识符、最新版本和默认版本。

以下示例提供指定启动模板的汇总。

```
aws ec2 describe-launch-templates --launch-template-names my-template-for-auto-scaling
```

以下为响应示例。

```
{
    "LaunchTemplates": [
        {
            "LaunchTemplateId": "lt-068f72b729example",
            "LaunchTemplateName": "my-template-for-auto-scaling",
            "CreateTime": "2020-02-28T19:52:27.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersionNumber": 1,
            "LatestVersionNumber": 1
        }
    ]
}
```

如果您未使用 `--launch-template-names` 选项将输出限制为一个启动模板，则将返回所有启动模板的相关信息。

以下 describe-launch-template-versions 命令提供描述指定启动模板版本的信息。

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b729example
```

以下为响应示例。

```
{
    "LaunchTemplateVersions": [
        {
            "VersionDescription": "version1",
            "LaunchTemplateId": "lt-068f72b729example",
            "LaunchTemplateName": "my-template-for-auto-scaling",
            "VersionNumber": 1,
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "LaunchTemplateData": {
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "purpose",
                                "Value": "webserver"
                            }
                        ]
                    }
                ],
                "ImageId": "ami-04d5cc9b88example",
                "InstanceType": "t2.micro",
                "NetworkInterfaces": [
                    {
                        "DeviceIndex": 0,
                        "DeleteOnTermination": true,
                        "Groups": [
                            "sg-903004f88example"
                        ],
                        "AssociatePublicIpAddress": true
```

```
                }
            ]
        },
        "DefaultVersion": true,
        "CreateTime": "2020-02-28T19:52:27.000Z"
    }
]
}
```

创建启动模板版本

以下 create-launch-template-version 命令将根据启动模板的版本 1 创建新的启动模板版本并指定其他 AMI ID。

```
aws ec2 create-launch-template-version --launch-template-id lt-068f72b729example --version-
description version2 \
  --source-version 1 --launch-template-data "ImageId=ami-c998b6b2example"
```

要设置启动模板的默认版本，请使用 modify-launch-template 命令。

删除启动模板版本

以下 delete-launch-template-versions 命令将删除指定的启动模板版本。

```
aws ec2 delete-launch-template-versions --launch-template-id lt-068f72b729example --
versions 1
```

删除启动模板

如果不再需要某个启动模板，则可以使用以下 delete-launch-template 命令进行删除。如果删除启动模板，则会删除该模板的所有版本。

```
aws ec2 delete-launch-template --launch-template-id lt-068f72b729example
```

更新 Auto Scaling 组以使用启动模板

您可以使用 update-auto-scaling-group 命令将启动模板添加到现有 Auto Scaling 组。

> **Note**
>
> 如果您将 Auto Scaling 组切换为使用启动配置，请确保您的权限是最新的。为了使用启动模板，您需要特定的权限。

更新 Auto Scaling 组以使用最新版本的启动模板

以下 update-auto-scaling-group 命令会更新指定的 Auto Scaling 组以使用最新版本的指定启动模板。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateId=lt-068f72b729example,Version='$Latest'
```

更新 Auto Scaling 组以使用特定版本的启动模板

以下 update-auto-scaling-group 命令会更新指定的 Auto Scaling 组以使用特定版本的指定启动模板。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

在启动模板中使用 Amazon Systems Manager 参数代替 AMI ID

Amazon EC2 Auto Scaling 支持在启动模板中使用引用了亚马逊云机器镜像（AMI）ID 的 Amazon Systems Manager 参数。借助 Systems Manager 参数，您可以更新您的自动扩缩组以使用新的 AMI ID，而无需在每次 AMI ID 更改时创建新的启动模板或启动模板的新版本。这些 ID 可能会定期更改，例如当 AMI 使用最新的操作系统或软件更新进行更新时。

您可以使用 Parameter Store（一项 Amazon Systems Manager 功能）创建、更新或删除 Systems Manager 参数。必须先创建 Systems Manager 参数，然后才能在启动模板中使用该参数。首先，您可以创建数据类型为 `aws:ec2:image` 的参数，并输入 AMI ID 作为其值。AMI ID 的格式为 `ami-<identifier>`，例如，`ami-123example456`。AMI ID 是否正确取决于您在其中启动自动扩缩组的实例类型和 Amazon Web Services 区域。

> **Important**
>
> 要在启动模板中指定 Systems Manager 参数，您必须有权使用 `ssm:GetParameters` 操作。您还需要权限才能使用 `ssm:GetParameters` 操作来利用指定 Systems Manager 参数的启动模板。此举允许验证参数值。例如 IAM policy，请参阅《Amazon Systems Manager 用户指南》中的使用 IAM policy 限制对 Systems Manager 参数的访问。

有关更多信息，请参阅以下资源：

- 要在控制台中创建参数，请参阅《Amazon Systems Manager 用户指南》中的创建 Systems Manager 参数（控制台）。
- 要使用 Amazon CLI 创建参数，请参阅《Amazon Systems Manager 用户指南》中的创建 Systems Manager 参数（Amazon CLI）。
- 要创建参数版本和标签，请参阅《Amazon Systems Manager 用户指南》中的使用参数版本和使用参数标签。
- 有关确保您输入的参数值是有效的 AMI ID 的信息，请参阅《Amazon Systems Manager 用户指南》中的亚马逊云机器镜像 ID 的原生参数支持。
- 有关在启动模板中使用引用 AMI ID 的参数的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用 Systems Manager 参数代替 AMI ID。
- 您还可以在启动模板中为 Amazon 维护的公有 AMI 引用 Amazon Systems Manager 公有参数。要了解如何引用这些公有参数，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用 Systems Manager 查找最新的 Amazon Linux AMI 和《适用于 Windows 实例的 Amazon EC2 用户指南》中的使用 Systems Manager 查找最新的 Amazon Linux AMI。

Console

使用 Amazon Systems Manager 参数创建启动模板

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择启动模板，然后选择创建启动模板。
3. 对于设备模板名称，请为您的启动模板输入描述性名称。
4. 在 Application and OS Images (Amazon Machine Image) [应用程序和操作系统镜像（Amazon 机器映像）] 下，选择 Browse more AMIs（浏览其他 AMI）。
5. 选择搜索栏右侧的箭头按钮，然后选择指定自定义值/Systems Manager 参数。
6. 在指定自定义值或 Systems Manager 参数对话框中，执行以下操作：

   a. 对于 AMI ID 或 Systems Manager 参数字符串，使用以下格式之一输入 Systems Manager 参数名称：

      - **`resolve:ssm:parameter-name`**

- **resolve:ssm:*parameter-name*:*version-number***
- **resolve:ssm:*parameter-name*:*label***
- **resolve:ssm:*public-parameter***

b. 选择 Save（保存）。

7. 根据需要指定任何其他启动模板参数，然后选择创建启动模板。

有关从控制台创建启动模板的更多信息，请参阅 为 Auto Scaling 组创建启动模板 (p. 20)。

Amazon CLI

指定客户自有参数的示例启动模板。

1. 使用以下语法：`resolve:ssm:`*`parameter-name`*，其中 `resolve:ssm` 是标准前缀，*`parameter-name`* 是 Systems Manager 参数名称。

   以下示例可创建一个从名为 *`golden-ami`* 的现有 Systems Manager 参数获取 AMI ID 的启动模板。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling
 \
  --launch-template-data file://config.json
```

   config.json 的内容：

```
{
    "ImageId":"resolve:ssm:golden-ami",
    "InstanceType":"t2.micro"
}
```

   如果未指定，则参数的默认版本为最新版本。

   以下示例引用 golden-ami 参数的特定版本。该示例使用 golden-ami 参数的版本 3，但您可以使用任何有效的版本号。

```
{
    "ImageId":"resolve:ssm:golden-ami:3",
    "InstanceType":"t2.micro"
}
```

   以下类似示例引用了映射到 golden-ami 参数特定版本的参数标签 prod。

```
{
    "ImageId":"resolve:ssm:golden-ami:prod",
    "InstanceType":"t2.micro"
}
```

   下面是示例输出。

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-068f72b724example",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "CreateTime": "2022-12-27T17:11:21.000Z",
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
```

```
    }
}
```

2. 要确认启动模板获得的 AMI ID 是否为首选，请使用 describe-launch-template-versions 命令。该命令使用 `--resolve-alias` 选项将参数解析为实际的 AMI ID。

```
aws ec2 describe-launch-template-versions --launch-template-name my-template-for-auto-scaling \
   --versions $Default --resolve-alias
```

该示例返回 `ImageId` 的 AMI ID。使用此启动模板启动实例时，AMI ID 解析为 `ami-04d5cc9b88example`。

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-068f72b724example",
            "LaunchTemplateName": "my-template-for-auto-scaling",
            "VersionNumber": 1,
            "CreateTime": "2022-12-27T17:11:21.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-04d5cc9b88example",
                "InstanceType": "t2.micro"
            }
        }
    ]
}
```

指定 Amazon 自有的公有参数示例启动模板

1. 请使用以下语法：`resolve:ssm:public-parameter`，其中 `resolve:ssm` 是标准前缀，*public-parameter* 是公有参数的路径和名称。

在此示例中，启动模板使用 Amazon 提供的公有参数，使用在 Amazon Web Services 区域 中为您的配置文件配置的最新 Amazon Linux 2 AMI 启动实例。

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling
 --version-description version1 \
  --launch-template-data file://config.json
```

`config.json` 的内容：

```
{
    "ImageId":"resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
    "InstanceType":"t2.micro"
}
```

以下为响应示例。

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-089c023a30example",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "CreateTime": "2022-12-28T19:52:27.000Z",
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
```

```
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

2. 要确认启动模板获得的 AMI ID 是否正确，请使用 describe-launch-template-versions 命令。该命令使用 `--resolve-alias` 选项将参数解析为实际的 AMI ID。

```
aws ec2 describe-launch-template-versions --launch-template-name my-template-for-auto-scaling \
   --versions $Default --resolve-alias
```

该示例返回 `ImageId` 的 AMI ID。使用此启动模板启动实例时，AMI ID 解析为 `ami-0ac394d6a3example`。

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-089c023a30example",
            "LaunchTemplateName": "my-template-for-auto-scaling",
            "VersionNumber": 1,
            "CreateTime": "2022-12-28T19:52:27.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0ac394d6a3example",
                "InstanceType": "t2.micro",
            }
        }
    ]
}
```

限制

- 目前，您不能在具有混合实例策略 (p. 50)的自动扩缩组上使用指定 Systems Manager 参数而不是 AMI ID 的启动模板。
- 您创建的 Systems Manager Parameter Store 参数必须存在于运行实例的同一账户中。
- Amazon EC2 Auto Scaling 仅支持将 AMI ID 指定为参数。
- 虽然您的自动扩缩组的启动模板引用了参数而不是 AMI ID，但您无法启动指定所需配置的实例刷新，也无法使用跳过匹配来替换组中的实例。
- 每次调用创建或更新自动扩缩组时，Amazon EC2 Auto Scaling 都会解析启动模板中的 Systems Manager 参数。如果您使用的是高级参数或更高的吞吐量限制，则频繁调用 Parameter Store（即 `GetParameters` 操作）可能会增加 Systems Manager 的成本，因为每次 Parameter Store API 交互都会产生费用。有关更多信息，请参阅 Amazon Systems Manager 定价。

启动配置

Important

我们强烈建议您不要使用启动配置。它们没有为 Amazon EC2 Auto Scaling 或 Amazon EC2 提供完整的功能。
启动配置不再添加对 2022 年 12 月 31 日之后发布的新 Amazon EC2 实例类型的支持。此外，2023 年 3 月 31 日之后创建的任何新账户都无法选择通过控制台创建新的启动配置。但是，API 和 CLI 访问权限将适用于在 2023 年 3 月 31 日到 2023 年 12 月 31 日之间创建的新账户，以支持客户使用自动化使用案例。2023 年 12 月 31 日之后创建的新账户将无法使用控制台、API 或 CLI 创建新的启动配置。有关将您的启动配置转换为启动模板的信息，请参阅 Amazon 计算博客上的 Amazon EC2 Auto Scaling 将不再向启动配置添加对新 EC2 功能的支持。

启动配置是 Auto Scaling 组用于启动 EC2 实例的实例配置模板。在创建启动配置时，您需要指定实例的信息。包括 Amazon Machine Image (AMI) 的 ID、实例类型、密钥对、一个或多个安全组以及块储存设备映射。如果之前已启动过 EC2 实例，可以指定相同的信息来启动实例。

您可以为多个 Auto Scaling 组指定启动配置。但是一次只能为一个 Auto Scaling 组指定一个启动配置，而且启动配置在创建后不能修改。要更改 Auto Scaling 组的启动配置，必须创建启动配置，然后用该配置更新您的 Auto Scaling 组。

请记住，只要在您创建或更新 Auto Scaling 组时，就必须指定启动配置、启动模板或 EC2 实例。当您使用 EC2 实例创建 Auto Scaling 组时，Amazon EC2 Auto Scaling 将自动为您创建启动配置并将其与 Auto Scaling 组关联。有关更多信息，请参阅 使用现有实例的参数创建 Auto Scaling 组 (p. 81)。或者，如果您使用启动模板，您可以指定启动模板而非启动配置或 EC2 实例。有关更多信息，请参阅启动模板 (p. 19)。

目录

- 创建启动配置 (p. 39)
- 使用 EC2 实例创建启动配置 (p. 42)
- 更改 Auto Scaling 组的启动配置 (p. 46)
- 使用启动配置来配置实例租赁 (p. 46)

创建启动配置

Important

我们强烈建议您不要使用启动配置。它们没有为 Amazon EC2 Auto Scaling 或 Amazon EC2 提供完整的功能。
启动配置不再添加对 2022 年 12 月 31 日之后发布的新 Amazon EC2 实例类型的支持。此外，2023 年 3 月 31 日之后创建的任何新账户都无法选择通过控制台创建新的启动配置。但是，API 和 CLI 访问权限将适用于在 2023 年 3 月 31 日到 2023 年 12 月 31 日之间创建的新账户，以支持客户使用自动化使用案例。2023 年 12 月 31 日之后创建的新账户将无法使用控制台、API 或 CLI 创建新的启动配置。有关将您的启动配置转换为启动模板的信息，请参阅 Amazon 计算博客上的 Amazon EC2 Auto Scaling 将不再向启动配置添加对新 EC2 功能的支持。

在创建启动配置时，您必须指定有关要启动的 EC2 实例的信息。包括 Amazon Machine Image (AMI)、实例类型、密钥对、安全组和块储存设备映射。或者，您可以使用正在运行的 EC2 实例的属性来创建启动配置。有关更多信息，请参阅使用 EC2 实例创建启动配置 (p. 42)。

在创建启动配置后，您可创建 Auto Scaling 组。有关更多信息，请参阅使用启动配置创建 Auto Scaling 组 (p. 80)。

Auto Scaling 组一次与一个启动配置关联，并且启动配置在创建后不能修改。因此，如果您要更改现有 Auto Scaling 组的启动配置，则必须使用新的启动配置更新它。有关更多信息，请参阅更改 Auto Scaling 组的启动配置 (p. 46)。

目录

- 创建启动配置（控制台）(p. 40)
- 创建启动配置（Amazon CLI）(p. 41)
- 配置实例元数据选项 (p. 41)

创建启动配置（控制台）

创建启动配置（控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的 Auto Scaling (自动调整) 上，选择 Launch Configurations (启动配置)。
3. 在导航栏中，选择 Amazon 区域。
4. 选择创建启动配置，然后为您的启动配置输入名称。
5. 对于 Amazon Machine Image (AMI)，选择 AMI。要查找特定 AMI，您可以查找合适的 AMI，记下其 ID，然后输入 ID 作为搜索条件。

   要获取 Amazon Linux 2 AMI 的 ID，请执行以下操作：

   a. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
   b. 在导航窗格中的实例下，选择实例，然后选择启动实例。
   c. 在选择 Amazon Machine Image 页面的 Quick Start 选项卡上，请注意 Amazon Linux 2 AMI (HVM) 旁边的 AMI 的 ID。
6. 对于实例类型，为您的实例选择硬件配置。
7. 在其他配置下，请注意以下字段：

   a. （可选）对于购买选项，您可以选择请求竞价型实例以按需价格为上限的 Spot 价格请求竞价型实例。（可选）您可以指定您的竞价型实例的每实例小时最高价。

      Note

      如果您能灵活控制应用程序的运行时间并且应用程序可以中断，那么相对于按需型实例，竞价型实例是经济实惠之选。有关更多信息，请参阅 为容错和灵活的应用程序请求竞价型实例 (p. 28)。

   b. （可选）对于 IAM 实例配置文件，选择要与实例关联的角色。有关更多信息，请参阅 在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)。
   c. （可选）对于监控，选择是否启用实例，通过启用详细监控能够以 1 分钟的间隔向 Amazon CloudWatch 发布指标数据。将收取额外费用。有关更多信息，请参阅 配置 Auto Scaling 实例的监控 (p. 253)。
   d. （可选）对于高级详细信息、用户数据，您可以指定用户数据在启动过程中配置实例或在实例启动后运行配置脚本。
   e. （可选）对于高级详细信息、IP 地址类型，选择是否将公有 IP 地址分配到组的实例。如果未设置值，则原定设置为使用启动实例的子网的自动分配公有 IP 设置。
8. （可选）对于存储（卷），如果您不需要额外存储，则可以跳过此部分。否则，除了 AMI 指定的卷以外，要指定要附加到实例的卷，请选择添加新卷。然后选择所需的选项和设备、快照、大小、卷类型、IOPS、吞吐量、终止时删除和已加密的关联值。
9. 对于安全组，创建或选择要与组实例关联的安全组。如果您保留选中创建新安全组选项，将为运行 Linux 的 Amazon EC2 实例配置默认 SSH 规则。将为运行 Windows 的 Amazon EC2 实例配置默认 RDP 角色。

10. 对于密钥对（登录），请选择密钥对选项下的选项。

    如果您已配置 Amazon EC2 实例密钥对，则可在此处选择它。

    如果您尚未拥有 Amazon EC2 实例密钥对，请选择创建新密钥对并为密钥对指定一个可识别的名称。选择下载密钥对以将密钥对下载到您的计算机。

    Important

    如果需要连接到您的实例，不要选择在没有密钥对的情况下继续。
11. 选中确认复选框，然后选择 Create launch configuration（创建启动配置）。

创建启动配置（Amazon CLI）

使用命令行创建启动配置

您可以使用以下任一命令：

- create-launch-configuration (Amazon CLI)
- New-ASLaunchConfiguration (Amazon Tools for Windows PowerShell)

配置实例元数据选项

Amazon EC2 Auto Scaling 支持在启动配置中配置实例元数据服务 (IMDS)。这样，您可以选择使用启动配置将 Auto Scaling 组中的 Amazon EC2 实例配置为要求实例元数据服务版本 2 (IMDSv2)，这是一种面向会话的方法，用于请求实例元数据。有关 IMDSv2 优势的详细信息，请参阅本文 Amazon 博客上有关为 EC2 实例元数据服务增加深度防御的增强功能的信息。

您可以将 IMDS 配置为同时支持 IMDSv2 和 IMDSv1（原定设置），或者要求使用 IMDSv2。如果您使用 Amazon CLI 或其中一个 SDK 配置 IMDS，必须使用最新版本的 Amazon CLI 或 SDK 以要求使用 IMDSv2。

您可以针对以下内容配置启动配置：

- 在请求实例元数据时要求使用 IMDSv2
- 指定 PUT 响应跃点限制
- 关闭对实例元数据的访问

您可以在以下主题中找到有关配置实例元数据服务的更多详细信息：适用于 Linux 实例的 Amazon EC2 用户指南中的配置实例元数据服务。

使用以下步骤在启动配置中配置 IMDS 选项。创建启动配置后，您可以将其与 Auto Scaling 组关联。如果将启动配置与现有 Auto Scaling 组关联，现有启动配置将与 Auto Scaling 组取消关联，并且现有实例需要替换才能使用您在新启动配置中指定的 IMDS 选项。有关更多信息，请参阅更改 Auto Scaling 组的启动配置 (p. 46)。

在启动配置中配置 IMDS（控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的 Auto Scaling (自动调整) 上，选择 Launch Configurations (启动配置)。
3. 在导航栏中，选择 Amazon 区域。
4. 选择创建启动配置，然后按照常规方式创建启动配置。包括 Amazon Machine Image (AMI) 的 ID、实例类型和可选的密钥对、一个或多个安全组以及实例的任何其他EBS 卷或实例存储卷。

5. 要为与此启动配置关联的所有实例配置实例元数据选项，请在其他配置中的高级详细信息下，执行以下操作：

   a. 对于元数据可访问，选择是启用还是禁用对实例元数据服务的 HTTP 终端节点的访问。预设情况下，将启用 HTTP 终端节点。如果您选择禁用终端节点，则会关闭对实例元数据的访问。只有在启用 HTTP 终端节点时，才可以指定要求 IMDSv2 的条件。

   b. 对于元数据版本，您可以选择在请求实例元数据时要求使用实例元数据服务版本 2 (IMDSv2)。如果您不指定值，原定设置为同时支持 IMDSv1 和 IMDSv2。

   c. 对于元数据标记响应跃点限制，您可以为元数据标记设置允许的网络跃点数。如果您未指定值，则原定设置为 1。

6. 完成后，选择创建启动配置。

要求在启动配置中使用 Amazon CLI 使用 IMDSv2

在 `--metadata-options` 设置为 `HttpTokens=required` 的情况下使用以下 create-launch-configuration 命令。在为 `HttpTokens` 指定值时，还必须将 `HttpEndpoint` 设置为已启用。由于安全令牌标头设置为需要元数据检索请求，因此会在请求实例元数据时使实例选择要求 IMDSv2。

```
aws autoscaling create-launch-configuration \
  --launch-configuration-name my-lc-with-imdsv2 \
  --image-id ami-01e24be29428c15b2 \
  --instance-type t2.micro \
      ...
  --metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

关闭对实例元数据的访问

请使用以下 create-launch-configure 命令关闭对实例元数据的访问。您可以稍后使用 modify-instance-metadata-options 命令重新打开访问权限。

```
aws autoscaling create-launch-configuration \
  --launch-configuration-name my-lc-with-imds-disabled \
  --image-id ami-01e24be29428c15b2 \
  --instance-type t2.micro \
      ...
  --metadata-options "HttpEndpoint=disabled"
```

使用 EC2 实例创建启动配置

Amazon EC2 Auto Scaling 提供了一个选项，可使用正在运行的 EC2 实例中的属性创建启动配置。

如果指定的实例有启动配置当前不支持的属性，则由Auto Scaling 组启动的实例可能不同于原始 EC2 实例。

从头创建启动配置和从现有 EC2 实例创建启动配置之间存在差异。如果从头创建启动配置，需要指定映像 ID、实例类型、可选资源（如存储设备）和可选设置（如监控）。从运行的实例中创建启动配置时，Amazon EC2 Auto Scaling 从指定的实例中派生启动配置的属性。属性也可能来自从中启动实例的 AMI 的块储存设备映射，从而忽略在启动后添加的任何其他块储存设备。

如果使用正在运行的实例创建启动配置，可以通过将以下属性指定为相同请求的一部分，来覆盖这些属性：AMI、块储存设备、密钥对、实例配置文件、实例类型、内核、实例监控、部署租用、虚拟磁盘、安全组、最高 Spot 价格、用户数据、该实例是否有公有 IP 地址，以及该实例是否经过了 EBS 优化。

> Tip
>
> 您可以直接从 EC2 实例创建 Auto Scaling 组 (p. 81)。如果使用此功能，Amazon EC2 Auto Scaling 还自动为您创建启动配置。

以下示例介绍如何从 EC2 实例创建启动配置。

示例

- 使用 EC2 实例创建启动配置 (p. 43)
- 从实例创建启动配置以及覆盖块储存设备 (Amazon CLI) (p. 44)
- 创建启动配置和覆盖实例类型 (Amazon CLI) (p. 45)

使用 EC2 实例创建启动配置

要使用现有 EC2 实例的属性创建启动配置，请指定该实例的 ID。

> Important
>
> 用于启动指定实例的 AMI 必须仍然存在。

从 EC2 实例创建启动配置（控制台）

您可以使用控制台从运行中的 EC2 实例创建启动配置和 Auto Scaling 组并将该实例添加到新的 Auto Scaling 组。有关更多信息，请参阅将 EC2 实例附上 Auto Scaling 组。 (p. 125)。

从 EC2 实例创建启动配置 (Amazon CLI)

可以使用以下 create-launch-configuration 命令从一个实例中创建启动配置，它使用与该实例相同的属性。将忽略在启动后添加的任何块储存设备。

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-instance
 --instance-id i-a8e09d9c
```

您可以使用以下 describe-launch-configurations 命令来描述启动配置，以及验证其属性是否与实例的属性相匹配。

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-
instance
```

以下为响应示例。

```
{
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn",
            "InstanceMonitoring": {
                "Enabled": false
            },
            "ImageId": "ami-05355a6c",
            "CreatedTime": "2014-12-29T16:14:50.382Z",
            "BlockDeviceMappings": [],
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                "sg-8422d1eb"
            ],
            "LaunchConfigurationName": "my-lc-from-instance",
            "KernelId": "null",
            "RamdiskId": null,
            "InstanceType": "t1.micro",
            "AssociatePublicIpAddress": true
```

```
        }
    ]
}
```

从实例创建启动配置以及覆盖块储存设备 (Amazon CLI)

预设情况下，Amazon EC2 Auto Scaling 使用您指定的 EC2 实例中的属性创建启动配置。不过，块储存设备来自用于启动实例的 AMI，而不是来自实例。要将块储存设备添加到启动配置，请覆盖该启动配置的块储存设备映射。

Important

用于启动指定实例的 AMI 必须仍然存在。

创建启动配置和覆盖块储存设备

使用以下 create-launch-configuration 命令来创建使用 EC2 实例但具有自定义块储存设备映射的启动配置。

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-instance-bdm --instance-id i-a8e09d9c \
  --block-device-mappings "[{\"DeviceName\":\"/dev/sda1\",\"Ebs\":{\"SnapshotId\":\"snap-3decf207\"}},{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"SnapshotId\":\"snap-eed6ac86\"}}]"
```

使用以下 describe-launch-configurations 命令来描述启动配置，并且验证它是否使用您的自定义块储存设备映射。

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-instance-bdm
```

下面的示例响应描述了该启动配置。

```
{
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn",
            "InstanceMonitoring": {
                "Enabled": false
            },
            "ImageId": "ami-c49c0dac",
            "CreatedTime": "2015-01-07T14:51:26.065Z",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "SnapshotId": "snap-3decf207"
                    }
                },
                {
                    "DeviceName": "/dev/sdf",
                    "Ebs": {
                        "SnapshotId": "snap-eed6ac86"
                    }
                }
            ],
```

```
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                "sg-8637d3e3"
            ],
            "LaunchConfigurationName": "my-lc-from-instance-bdm",
            "KernelId": null,
            "RamdiskId": null,
            "InstanceType": "t1.micro",
            "AssociatePublicIpAddress": true
        }
    ]
}
```

创建启动配置和覆盖实例类型 (Amazon CLI)

预设情况下，Amazon EC2 Auto Scaling 使用您指定的 EC2 实例中的属性创建启动配置。根据您的要求，您可能需要覆盖实例中的属性并使用所需的值。例如，您可以覆盖实例类型。

Important

用于启动指定实例的 AMI 必须仍然存在。

创建启动配置和覆盖实例类型

使用以下 create-launch-configuration 命令来创建使用 EC2 实例但实例类型（例如 `t2.medium`）不同于该示例（例如 `t2.micro`）的启动配置。

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-instance-changetype \
  --instance-id i-a8e09d9c --instance-type t2.medium
```

使用以下 describe-launch-configurations 命令来描述启动配置以及验证实例类型是否被覆盖。

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-instance-changetype
```

下面的示例响应描述了该启动配置。

```
{
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn",
            "InstanceMonitoring": {
                "Enabled": false
            },
            "ImageId": "ami-05355a6c",
            "CreatedTime": "2014-12-29T16:14:50.382Z",
            "BlockDeviceMappings": [],
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                "sg-8422d1eb"
            ],
            "LaunchConfigurationName": "my-lc-from-instance-changetype",
            "KernelId": "null",
            "RamdiskId": null,
            "InstanceType": "t2.medium",
            "AssociatePublicIpAddress": true
        }
```

```
    ]
}
```

更改 Auto Scaling 组的启动配置

Auto Scaling 组一次与一个启动配置关联，并且启动配置在创建后不能修改。要更改 Auto Scaling 组的启动配置，请将现有的启动配置作为新启动配置的基础。然后，更新 Auto Scaling 组以使用新的启动配置。

更改 Auto Scaling 组的启动配置后，将使用新的配置选项启动所有新实例，但现有实例不受影响。要更新现有实例，请终止这些实例，以便 Auto Scaling 组替换这些实例，或者根据您的终止策略 (p. 222)启用自动扩展，以逐步使用较新实例替换较旧实例。

> **Note**
>
> 您还可以替换 Auto Scaling 组中的所有实例，以启动使用新启动配置的新实例。有关更多信息，请参阅 替换 Auto Scaling 实例 (p. 94)。

要更改 Auto Scaling 组的启动配置（控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的 Auto Scaling (自动调整) 上，选择 Launch Configurations (启动配置)。
3. 选择启动配置，选择 Actions（操作），然后单击 Copy launch configuration（复制启动配置）。这将设置与原启动配置选项相同的新启动配置，但在名称中会增加“Copy”文本。
4. 在 Copy Launch Configuration（复制启动配置）页面上，根据需要编辑配置选项，然后选择 Create launch configuration（创建启动配置）。
5. 在导航窗格的 Auto Scaling 下，选择 自动扩缩组。
6. 选中 Auto Scaling 组旁边的复选框。

   将在页面底部打开一个拆分窗格，其中显示有关所选组的信息。
7. 在详细信息选项卡上，选择启动配置、编辑。
8. 对于启动配置，选择新的启动配置。
9. 完成后，选择更新。

更改 Auto Scaling 组的启动配置 (Amazon CLI)

1. 使用 describe-launch-configurations 命令描述当前的启动配置。
2. 使用 create-launch-configuration 命令创建新的启动配置。
3. 带 `--launch-configuration-names` 参数使用 update-auto-scaling-group 命令更新 Auto Scaling 组的启动配置。

要更改 Auto Scaling 组的启动配置 (Tools for Windows PowerShell)

1. 使用 Get-ASLaunchConfiguration 命令描述当前的启动配置。
2. 使用 New-ASLaunchConfiguration 命令创建新的启动配置。
3. 带 `-LaunchConfigurationName` 参数使用 Update-ASAutoScalingGroup 命令更新 Auto Scaling 组的启动配置。

使用启动配置来配置实例租赁

租赁定义如何在物理硬件之间分配 EC2 实例及其对定价的影响。有三种租赁选项可供选择：

- 共享 (default) — 多个 Amazon Web Services 账户 可以共享相同物理硬件。
- 专用实例 (dedicated) — 您的实例在单个租户硬件上运行。
- 专用主机 (host) — 您的实例在一个 EC2 实例容量完全供您专用的物理服务器上运行，这是一个具有您可以控制的配置的隔离服务器。

本主题介绍如何通过在启动配置中指定设置来在 Auto Scaling 组中启动专用实例。有关定价信息和了解有关专用实例的更多信息，请参阅 Amazon EC2 专用实例产品页面和适用于 Linux 实例的 Amazon EC2 用户指南中的专用实例。

您可以使用启动配置或启动模板为 EC2 实例配置租赁。但是，host 租赁值不能与启动配置一起使用。仅使用 default 或 dedicated 租赁值。

> **Important**
>
> 要使用租赁值 host，您必须使用启动模板。有关更多信息，请参阅为 Auto Scaling 组创建启动模板 (p. 20)。启动专用主机之前，我们建议您熟悉使用 Amazon License Manager 启动和管理专用主机。有关更多信息，请参阅 License Manager 用户指南。

当您创建 VPC 时，默认情况下它的租期属性设置为 default。在这样一个 VPC 中，您可启动租期值为 dedicated 的实例，以便它们作为单一租期实例运行。否则，默认情况下，这些实例作为共享租期实例运行。如您将 VPC 的租期属性设置为 dedicated，则在 VPC 内启动的所有实例都作为单一租期实例运行。

在创建启动配置后，实例部署租期的默认值为 null，实例租期由 VPC 的租期属性控制。您可以使用带 --placement-tenancy 选项的 create-launch-configure CLI 命令将启动配置的实例部署租期指定为 default 或者 dedicated。

下表总结了在 VPC 中启动的 Auto Scaling 实例的实例部署租期。

启动配置租期	VPC 租期 = **default**	VPC 租期 = **dedicated**
未指定	共享租期实例	Dedicated Instances
default	共享租期实例	Dedicated Instances
dedicated	Dedicated Instances	Dedicated Instances

创建用于创建专用实例的启动配置 (Amazon CLI)

使用以下 create-launch-configure 命令创建将启动配置租期设置为 dedicated 的启动配置。

```
aws autoscaling create-launch-configuration --launch-configuration-name my-launch-config --
placement-tenancy dedicated --image-id ...
```

您可以使用以下 describe-launch-configurations 命令验证启动配置的实例部署租期。

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-launch-
config
```

以下是创建专用实例的启动配置的示例输出。只有在明确设置实例部署租期时，PlacementTenancy 参数才是该命令输出的一部分。

```
{
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
```

```
            "PlacementTenancy": "dedicated",
            "LaunchConfigurationARN": "arn",
            "InstanceMonitoring": {
                "Enabled": true
            },
            "ImageId": "ami-b5a7ea85",
            "CreatedTime": "2020-03-08T23:39:49.011Z",
            "BlockDeviceMappings": [],
            "KeyName": null,
            "SecurityGroups": [],
            "LaunchConfigurationName": "my-launch-config",
            "KernelId": null,
            "RamdiskId": null,
            "InstanceType": "m3.medium"
        }
    ]
}
```

自动扩缩组

> **Note**
>
> 如果您不熟悉自动扩缩组，可以在 Amazon EC2 Auto Scaling 入门 (p. 13) 教程中了解更多信息。首先创建启动模板或启动配置，然后使用它创建自动扩缩组，其中所有实例都具有相同的实例属性。例如，您可以通过将以下实例属性指定为启动模板或启动配置的一部分来设置这些属性：Amazon 机器映像（AMI）、实例类型、块储存设备、SSH 密钥对以及控制实例入站和出站流量的安全组。

自动扩缩组由一组 Amazon EC2 实例组成，这是实例视为一个逻辑组以实现自动扩缩和管理。另外，自动扩缩组让您能够使用 Amazon EC2 Auto Scaling 功能，如运行状况检查替换和扩展策略。保持 Auto Scaling 组中的实例数量和自动扩展都是 Amazon EC2 Auto Scaling 服务的核心功能。

Auto Scaling 组的大小取决于您设置为所需容量的实例数量。您可以通过手动方式或使用自动扩展调整其大小以满足需求。

Auto Scaling 组会首先启动足够实例以达到需要的容量。它通过对组中实例定期执行运行状况检查来保持实例数量。即使某个实例运行状况不佳，Auto Scaling 组也会继续保持固定数量的实例。如果某个实例运行状况不佳，则该组终止运行状况不佳的实例，并启动其他实例来替换它。有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。

您可以使用扩展策略动态增加或减少组中的实例数量，以符合不断变化的条件。扩展策略生效时，Auto Scaling 组会在您指定的最小和最大容量值之间调整组的所需容量，并根据需要启动或终止实例。您还可以按计划进行扩展。有关更多信息，请参阅扩展 Auto Scaling 组的大小 (p. 121)。

Auto Scaling 组可以启动按需型实例和/或竞价型实例。仅在您将自动扩缩组配置为使用启动模板时，才可以为该组指定多个购买选项。建议您使用启动模板而不是启动配置，以确保您可以使用 Amazon EC2 Auto Scaling 和 Amazon EC2 的最新功能。

竞价型实例让您能够以相对于按需价格而言极低的折扣，获取未使用的 EC2 容量。有关更多信息，请参阅 Amazon EC2 竞价型实例。竞价型实例和按需型实例之间存在着关键区别：

- 竞价型实例的价格因需求而异
- Amazon EC2 可以随着竞价型实例的可用性或价格发生变化而终止单个竞价型实例

竞价型实例终止后，Auto Scaling 组会尝试启动替代实例来保持该组的所需容量。

实例启动时，如果您指定多个可用区，会为这些可用区分配所需容量。如果执行扩展操作，Amazon EC2 Auto Scaling 会自动保持您指定的所有可用区的平衡。

目录

- Auto Scaling 组具有多个实例类型和购买选项 (p. 50)
- 示例：使用基于属性的实例类型选择创建 Auto Scaling 组 (p. 72)
- 使用启动模板创建 Auto Scaling 组 (p. 78)
- 使用启动配置创建 Auto Scaling 组 (p. 80)
- 使用现有实例的参数创建 Auto Scaling 组 (p. 81)
- 使用 Amazon EC2 启动向导创建 Auto Scaling 组 (p. 85)
- 为 Auto Scaling 组和实例添加标签 (p. 87)

- 替换 Auto Scaling 实例 (p. 94)
- 删除 Auto Scaling 基础设施 (p. 117)

Auto Scaling 组具有多个实例类型和购买选项

您可以启动并自动扩展单个 Auto Scaling 组中的一组按需实例和竞价型实例。除了享受使用竞价型实例的折扣外，您还可以使用预留实例或 Savings Plan 获得常规按需实例定价的折扣费率。以上所有因素的综合作用是帮助您进一步节约 EC2 实例成本，同时帮助您获得应用程序所需的规模和性能。

以下步骤将介绍如何创建自动扩缩组：

- 指定用于启动实例的启动模板。
- 选择要在其中启动您的自动扩缩组的 VPC 和子网。
- 选择使用新要求覆盖启动模板的现有实例类型要求的选项。
- 手动选择您的实例类型并确定其优先级。例如，您可以选择优先考虑可以从按需型实例的 Savings Plan 或预留实例折扣定价中受益的实例类型。
- 指定要启动的按需型实例和竞价型实例的百分比。
- 选择决定 Amazon EC2 Auto Scaling 将如何利用可能的实例类型来达到您的按需型和竞价型容量的分配策略。
- 指定组大小，包括所需容量、最小容量和最大容量。

通过跨多个可用区中运行的多个实例类型部署应用程序以提高可用性。虽然您可以使用一种实例类型，但最佳实践是使用多种实例类型。这样，如果您选择的可用区中实例容量不足，则 Amazon EC2 Auto Scaling 可以启动另一种实例类型。如果竞价型实例的实例容量不足，则 Amazon EC2 Auto Scaling 将继续尝试从其他竞价型实例池启动。（它使用的池由您选择的实例类型和分配策略决定。）Amazon EC2 Auto Scaling 通过启动竞价型实例而不是按需型实例，帮助您利用竞价型实例节省的成本。

目录

- 分配策略 (p. 50)
- 竞价型实例的最佳实践 (p. 52)
- 控制按需型实例的比例 (p. 52)
- 先决条件 (p. 54)
- 创建包括竞价型实例和按需型实例的 Auto Scaling 组（控制台）(p. 54)
- 创建包括竞价型实例和按需型实例的自动扩缩组（Amazon CLI）(p. 56)
- 验证您的自动扩缩组配置和已启动的实例 (Amazon CLI) (p. 63)
- 配置覆盖 (p. 63)

分配策略

以下分配策略决定 Auto Scaling 组如何根据可用的实例类型满足按需和竞价容量。

Amazon EC2 Auto Scaling 首先尝试确保实例在您指定的可用区之间均匀平衡。然后，它根据指定的分配策略启动实例类型。

竞价型实例

Amazon EC2 Auto Scaling 提供了以下可用于竞价型实例的分配策略：

`capacity-optimized`

Amazon EC2 Auto Scaling 将从池中请求您的竞价型实例，对于启动的实例数量而言，该池具有最佳容量。

使用竞价型实例，定价会根据长期供需趋势缓慢发生变化，但容量会实时波动。`capacity-optimized` 策略通过查看实时容量数据并预测可用性最高的池，自动在可用性最高的池中启动 Spot 实例。这有助于最大限度减少工作负载的可能中断，这些工作负载可能会因重启工作和检查点操作而导致更高的中断成本。要使某些实例类型有更高的首先启动机会，请使用 `capacity-optimized-prioritized`。

`capacity-optimized-prioritized`

设置启动模板覆盖的实例类型顺序，从最高优先级到最低优先级（从列表中的第一项到最后一项降序列出）。Amazon EC2 Auto Scaling 会尽最大努力遵守实例类型优先级，但首先会针对容量进行优化。对于必须最大限度地减少中断可能性，但对某些实例类型的偏好也很重要的工作负载来说，这是一个不错的选择。如果将按需型分配策略设置为 `prioritized`，则在满足按需型容量时将应用相同的优先级。

`lowest-price`

Amazon EC2 Auto Scaling 将在您为 Lowest priced pools (价格最低的池) 设置指定的 N 个竞价型池之间，使用可用区内价格最低的池请求您的竞价型实例。例如，如果您指定四种实例类型和四个可用区，则您的自动扩缩组可以访问最多 16 个竞价型池。（每个可用区内四个。）如果您为分配策略指定两个竞价型池 (N=2)，则您的自动扩缩组可以使用每个可用区的两个价格最低的池来满足竞价型容量。

由于此策略仅考虑实例价格而不考虑容量可用性，因此可能会导致较高的中断率。

`price-capacity-optimized`（推荐）

价格和容量优化分配策略同时考虑价格和容量，以选择中断可能性最小、价格尽可能低的竞价型实例池。

首先，建议您选择 `price-capacity-optimized` 分配策略并指定一组适合您的应用程序的实例类型。此外，您可以为 Amazon EC2 Auto Scaling 定义一系列可用区，以便在启动实例时从中选择。

有关 `price-capacity-optimized` 策略以及该策略能够提供帮助的应用场景的更多信息，请参阅 Amazon 博客中适用于 EC2 竞价型实例的价格容量优化分配策略的简介。

（可选）您可以指定您的竞价型实例的最高价。如果未指定最高价，则默认最高价为按需价格。但您仍然可以享受竞价型实例提供的大幅折扣。这些折扣得以实现，是由于借助竞价型定价模型带来了稳定的竞价型定价。

按需实例

Amazon EC2 Auto Scaling 提供了以下可用于按需实例的分配策略：

`lowest-price`

Amazon EC2 Auto Scaling 会根据当前的按需价格自动部署每个可用区中价格最低的实例类型。

为了满足您的所需容量，您可能会在每个可用区中获得一种以上实例类型的按需型实例。这取决于您请求多大容量。

`prioritized`

在满足按需型容量时，Amazon EC2 Auto Scaling 组将根据启动模板覆盖列表中实例类型的顺序，确定首先使用哪个实例类型。例如，假设您按以下顺序指定三个启动模板覆盖：`c5.large`、`c4.large` 和 `c3.large`。在您的按需型实例启动时，自动扩缩组满足按需型容量的顺序是从 `c5.large` 开始，然后是 `c4.large`，最后是 `c3.large`。

管理按需实例的优先级顺序时，请考虑以下事项：

您可以通过使用 Savings Plans 或预留实例预先支付使用费用，以获得按需实例的大幅折扣。有关更多信息，请参阅 Amazon EC2 定价页面。

- 对于预留实例，如果 Amazon EC2 Auto Scaling 启动匹配的实例类型，则适用常规按需型实例定价的折扣价。因此，如果存在未使用的 `c4.large` 预留实例，则可以设置实例类型优先级，将预留实例的最高优先级赋予 `c4.large` 实例类型。当 `c4.large` 实例启动时，您可享受预留实例定价。
- 通过 Savings Plans，当您使用 Amazon EC2 Instance Savings Plans 或 Compute Savings Plans 时，将适用常规按需型实例定价的折扣价。通过 Savings Plans，您可以更灵活地确定实例类型的优先级。只要您使用 Savings Plan 涵盖的实例类型，就可以按任何优先级顺序设置它们。您也可以偶尔更改实例类型的整个顺序，同时仍可享受 Savings Plan 折扣费率。有关 Savings Plans 的更多信息，请参阅 Savings Plans 用户指南。

竞价型实例的最佳实践

在创建 Auto Scaling 组以请求竞价型实例之前，请查看适用于 Linux 实例的 Amazon EC2 用户指南中的 EC2 竞价的最佳实践。使用这些最佳实践规划您的请求，以便以可能的最低价格预置需要的实例类型。还建议执行以下操作：

- 不要指定最高价格。如果您未收到任何竞价型实例 (例如当您的最高价太低时)，您的应用程序可能不运行。默认最高价格为按需型价格，您只需为您启动的竞价型实例支付竞价型价格。当竞价型价格低于最高价格时，是否满足您的请求取决于可用性。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的定价和节省。
- 一条很好的经验法则是，对于每种工作负载，灵活地在至少 10 种实例类型之间进行选择。此外，还要确保所有可用区配置为在 VPC 中使用，并为工作负载选择了这些可用区。由于可用区内每个实例类型的容量分开波动，因此，如果您灵活使用实例类型，则通常可以获得更多计算容量。
- 不要局限于最热门的新实例类型。选择老一代实例类型往往会导致中断较少，因为按需客户对它们的需求较少。
- 启用容量再平衡。执行此操作时，Amazon EC2 Auto Scaling 将尝试在 Amazon EC2 Spot 服务通知竞价型实例处于较高的中断风险时启动竞价型实例。启动新实例后，它会终止旧实例。有关更多信息，请参阅使用容量重新平衡来处理 Amazon EC2 竞价型实例中断 (p. 262)。
- 建议您使用 `price-capacity-optimized` 分配策略而不是 `lowest-price` 分配策略，以避免从中断可能性较高的池中请求竞价型实例。
- 如果您选择 `lowest-price` 分配策略并运行 Web 服务，则请指定较大的竞价型池数量，如 N=10。这样做可以减少其中一个可用区中的池暂时不可用时造成的竞价型实例中断的影响。如果您运行批处理或其他非关键任务应用程序，则可以指定较小竞价型池数量，例如，N=2。这样仅会从每个可用区中价格最低的竞价型池中预调配竞价型实例。

  Amazon EC2 Auto Scaling 会尽力尝试根据您指定的 N 个池提取竞价型实例。如果池在满足所需容量之前耗尽竞价容量，则 Amazon EC2 Auto Scaling 将继续通过从下一个价格最低的池中提取来满足您的请求。为了达到您所需的容量，您可能会从超过指定的 N 个池接收竞价型实例。同样，如果大多数池没有竞价容量，则您可能会从少于指定的 N 个池接收完整所需容量。

如果您打算指定最高价格，则可以使用 Amazon CLI 或 SDK 创建 Auto Scaling 组，但务必谨慎。如果您的最高价低于选定的实例类型的 Spot 价格，则不会启动您的竞价型实例。

控制按需型实例的比例

您可以完全控制 Auto Scaling 组中作为按需实例启动的实例所占比例。为确保您始终拥有实例容量，您可以指定组中作为按需型实例启动的百分比。或者，您也可以指定要开始使用的按需型实例的基本数量。如果您指定按需型实例的基本容量，则 Amazon EC2 Auto Scaling 将等待，直到它在组横向扩展时启动按需型实例的基本容量之后，才会启动竞价型实例。超出基本容量的任何内容都使用按需百分比来确定要启动多少按需型实例和竞价型实例。您可以指定 0 到 100 之间的任意数字来作为按需百分比。

Amazon EC2 Auto Scaling 将百分比转换为等效数量的实例。如果结果创建一个小数，Amazon EC2 Auto Scaling 向上舍入为下一个整数，以支持按需实例。

下表展示了自动扩缩组随着规模的增加而采取的行为。

示例：扩展行为

实例分配	各购买选项的运行实例总数			
	10	20	30	40
示例 1				
On-Demand base: 10	10	10	10	10
On-Demand percentage above base: 50%	0	5	10	15
Spot percentage: 50%	0	5	10	15
示例 2				
On-Demand base: 0	0	0	0	0
On-Demand percentage above base: 0%	0	0	0	0
Spot percentage: 100%	10	20	30	40
示例 3				
On-Demand base: 0	0	0	0	0
On-Demand percentage above base: 60%	6	12	18	24
Spot percentage: 40%	4	8	12	16
示例 4。				
On-Demand base: 0	0	0	0	0
On-Demand percentage above base: 100%	10	20	30	40
Spot percentage: 0%	0	0	0	0
示例 5				
On-Demand base: 12	10	12	12	12

实例分配	各购买选项的运行实例总数			
On-Demand percentage above base: 0%	0	0	0	0
Spot percentage: 100%	0	8	18	28

先决条件

创建启动模板，其中包括启动 EC2 实例所需的参数，例如 Amazon Machine Image (AMI) 和安全组。有关更多信息，请参阅为 Auto Scaling 组创建启动模板 (p. 20)。

验证启动模板尚未请求竞价型实例。

验证您拥有使用启动模板所需的权限。当您使用启动模板时，将检查您的 `ec2:RunInstances` 权限。如果启动模板指定了 IAM 角色，也会检查您的 `iam:PassRole` 权限。有关更多信息，请参阅启动模板支持 (p. 328)。

创建包括竞价型实例和按需型实例的 Auto Scaling 组（控制台）

完成以下步骤以创建您可以扩展的竞价型实例和按需型实例队组。

使用以下过程选择您的组可以启动的单个实例类型。如果您希望使用实例属性作为选择实例类型的标准，请通过指定您需要的 vCPU 数量和内存来配置实例类型要求。有关更多信息，请参阅示例：使用基于属性的实例类型选择创建 Auto Scaling 组 (p. 72)。您还可以指定存储类型、网络接口、CPU 制造商和加速器类型等其他属性。

创建包括竞价型实例和按需实例的 Auto Scaling 组

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 在屏幕顶部的导航栏中，选择在创建启动模板时使用的同一 Amazon Web Services 区域。
3. 选择 Create an Auto Scaling group (创建 Auto Scaling 组)。
4. 在选择启动模板或配置页面上，对于 Auto Scaling 组名称，输入 Auto Scaling 组的名称。
5. 要选择启动模板，请执行以下操作：
   a. 对于启动模板，请选择现有启动模板。
   b. 对于 Launch template version（启动模板版本），选择 Auto Scaling 组在扩展时使用启动模板的默认版本、最新版本还是特定版本。
   c. 验证您的启动模板是否支持您计划使用的所有选项，然后选择 Next（下一步）。
6. 要选择要在其中启动实例的 VPC 和子网，请执行以下操作：
   a. 在 Choose instance launch options (选择实例启动选项) 页面的 Network (网络) 下方，对于 VPC，选择相应的 VPC。Auto Scaling 组必须与您在启动模板中指定的安全组创建在相同的 VPC 中。
   b. 对于 Availability Zones and subnets (可用区和子网)，选择指定 VPC 中的一个或多个子网。可以在多个可用区中使用子网以提供高可用性。有关更多信息，请参阅选择 VPC 子网时的注意事项 (p. 302)。
7. 要手动选择您的组可以启动的单个实例类型，请执行以下操作：
   a. 对于 Instance type requirements（实例类型要求），选择 Override launch template（覆盖启动模板），然后选择 Manually add instance types（手动添加实例类型）。

b. 选择您的实例类型。您可以使用我们的建议作为起点。默认情况下，将选择 Family and generation flexible（系列和世代灵活）。

要更改实例类型的顺序，请使用箭头。如果您选择支持优先级排序的分配策略，则实例类型顺序将设置其启动优先级。

要删除实例类型，请选择 X。

或者，对于 Weight（权重）列中的方框，为每个实例类型分配一个相对权重。要执行此操作，请输入该类型的实例计入组所需容量的单元数量。如果实例类型提供不同的 vCPU、内存、存储或网络带宽功能，则执行此操作可能很有用。有关更多信息，请参阅为 Amazon EC2 Auto Scaling 配置实例权重 (p. 67)。

Note

如果您选择使用 Size flexible（大小灵活）建议，则属于本部分的所有实例类型都将自动获得权重值。如果您不想指定任何权重，则请清除 Weight（权重）列中所有实例类型的方框。

c. 在 Instance purchase options（实例购买选项）下，对于 Instances distribution（实例分配），分别指定要启动的按需型实例和竞价型实例在该组中所占的百分比。如果您的应用程序无状态、容错，并且可以处理中断的实例，则可以指定更高的竞价型实例百分比。

d. 如果您指定了一个竞价型实例百分比，则可以选中 Include On-Demand base capacity（包括按需基本容量）旁边的复选框，然后指定按需型实型例在自动扩缩组的最小初始容量中必须达到的数量。超出基本容量的任何内容都使用实例分配设置来确定要启动多少按需实例和 Spot 实例。

e. 在 Allocation strategies (分配策略) 下，对于 On-Demand allocation strategy (按需分配策略)，选择分配策略。当您手动选择实例类型时，默认情况下将选择 Prioritized（优先）。

f. 对于 Spot allocation strategy（竞价型分配策略），选择分配策略。默认情况下，将选择 Price capacity optimized（价格容量优化）。默认情况下，将隐藏 Lowest price（最低价格），仅在您选择 Show all strategies（显示所有策略）时才会出现。

Note

如果您选择了 Lowest price（最低价格），则请输入价格最低的池的数量，以跨价格最低的池实现多样化。如果您选择了 Capacity optimized（容量优化），则可以选择选中 Prioritize instance types（确定实例类型的优先级）复选框，以便让 Amazon EC2 Auto Scaling 根据您的实例类型列出的顺序选择首先启动哪个实例类型。

g. 对于 Capacity rebalance (容量再平衡)，选择启用还是禁用容量再平衡。

如果您选择了一个竞价型实例百分比，则可以使用容量再平衡功能，以自动响应竞价型实例因竞价中断而即将终止的情况。有关更多信息，请参阅使用容量重新平衡来处理 Amazon EC2 竞价型实例中断 (p. 262)。

h. 完成之后，选择 Next（下一步）两次，以转至 Configure group size and scaling policies（配置组大小和扩缩策略）页面。

8. 对于 Configure group size and scaling policies（配置组大小和扩缩策略）步骤，请执行以下操作：

a. 为 Desired capacity（所需容量）输入您的 Auto Scaling 组的初始大小，然后根据需要更新 Minimum capacity（最小容量）和 Maximum capacity（最大容量）。有关更多信息，请参阅为 Auto Scaling 组设置容量限制 (p. 122)。

Note

默认情况下，所需容量以及最小和最大容量限制将以实例数量表示。如果您为实例类型分配了权重，则必须将这些值转换为用于分配权重的相同度量单位，如 vCPU 的数量。

b. （可选）将组配置为通过指定目标跟踪扩展策略进行扩展。或者，在您创建组后指定此策略。有关更多信息，请参阅Amazon EC2 Auto Scaling 的目标跟踪扩缩策略 (p. 133)。

c. （可选）启用实例横向缩减保护功能，该功能可以防止您的自动扩缩组在横向缩减时终止实例。有关更多信息，请参阅实例横向缩减保护 (p. 230)。

d. 在完成后，选择下一步。

9. （可选）要在自动扩缩组扩展时收到通知，请为 Add notification（添加通知）配置通知，然后选择 Next（下一步）。有关更多信息，请参阅在 Auto Scaling 组扩展时获取 Amazon SNS 通知 (p. 258)。
10. （可选）要添加标签，请选择 Add tag（添加标签），为每个标签提供标签键和值，然后选择 Next（下一步）。有关更多信息，请参阅为 Auto Scaling 组和实例添加标签 (p. 87)。
11. 在 Review（查看）页面上，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

创建包括竞价型实例和按需型实例的自动扩缩组（Amazon CLI）

以下示例配置显示如何使用不同的竞价分配策略启动竞价型实例。

Note

这些示例显示如何使用 JSON 或 YAML 格式的配置文件。如果您使用 Amazon CLI 版本 1，必须指定 JSON 格式的配置文件。如果您使用 Amazon CLI 版本 2，可以指定 YAML 或 JSON 格式的配置文件。

示例

- 示例 1：使用 capacity-optimized 分配策略启动竞价型实例 (p. 56)
- 示例 2：使用 capacity-optimized-prioritized 分配策略启动竞价型实例 (p. 58)
- 示例 3：使用在两个池之间不同的 lowest-price 分配策略启动竞价型实例 (p. 59)
- 示例 4：使用 price-capacity-optimized 分配策略启动 Spot 实例 (p. 61)

示例 1：使用 capacity-optimized 分配策略启动竞价型实例

以下 create-auto-scaling-group 命令会创建指定以下内容的 Auto Scaling 组：

- 作为按需型实例 (0) 启动的组的百分比以及以 (1) 开头的按需实例的基本数量。
- 按优先级顺序（c5.large、c5a.large、m5.large、m5a.large、c4.large、m4.large、c3.large、m3.large）启动的实例类型。
- 要在其中启动实例的子网（subnet-5ea0c127、subnet-6194ea3b、subnet-c934b782）。每个子网都对应不同的可用区。
- 启动模板 (my-launch-template) 和启动模板版本 ($Default)。

Amazon EC2 Auto Scaling 尝试满足您的按需容量时，它会首先启动 c5.large 实例类型。竞价型实例来自每个可用区中基于竞价型实例容量的最佳竞价池。

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

下面是一个 config.json 示例文件。

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Default"
```

```
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandBaseCapacity": 1,
            "OnDemandPercentageAboveBaseCapacity": 0,
            "SpotAllocationStrategy": "capacity-optimized"
        }
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 3,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

或者，您可以使用以下 create-auto-scaling-group 命令创建自动扩缩组。这将引用 YAML 文件作为自动扩缩组的唯一参数，而不是 JSON 文件。

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

下面是一个 config.yaml 示例文件。

```
---
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
```

```
    - InstanceType: c3.large
    - InstanceType: m3.large
  InstancesDistribution:
    OnDemandBaseCapacity: 1
    OnDemandPercentageAboveBaseCapacity: 0
    SpotAllocationStrategy: capacity-optimized
MinSize: 1
MaxSize: 5
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

示例 2：使用 capacity-optimized-prioritized 分配策略启动竞价型实例

以下 create-auto-scaling-group 命令会创建指定以下内容的 Auto Scaling 组：

- 作为按需型实例 (0) 启动的组的百分比以及以 (1) 开头的按需实例的基本数量。
- 按优先级顺序（c5.large、c5a.large、m5.large、m5a.large、c4.large、m4.large、c3.large、m3.large）启动的实例类型。
- 要在其中启动实例的子网（subnet-5ea0c127、subnet-6194ea3b、subnet-c934b782）。每个子网都对应不同的可用区。
- 启动模板 (my-launch-template) 和启动模板版本 ($Latest)。

Amazon EC2 Auto Scaling 尝试满足您的按需容量时，它会首先启动 c5.large 实例类型。当 Amazon EC2 Auto Scaling 尝试满足您的竞价容量时，它会尽最大努力遵守实例类型优先级。但它首先会针对容量进行优化。

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

下面是一个 config.json 示例文件。

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
```

```
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandBaseCapacity": 1,
            "OnDemandPercentageAboveBaseCapacity": 0,
            "SpotAllocationStrategy": "capacity-optimized-prioritized"
        }
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 3,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

或者，您可以使用以下 create-auto-scaling-group 命令创建自动扩缩组。这将引用 YAML 文件作为自动扩缩组的唯一参数，而不是 JSON 文件。

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

下面是一个 config.yaml 示例文件。

```
---
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
  InstancesDistribution:
    OnDemandBaseCapacity: 1
    OnDemandPercentageAboveBaseCapacity: 0
    SpotAllocationStrategy: capacity-optimized-prioritized
MinSize: 1
MaxSize: 5
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

示例 3：使用在两个池之间不同的 lowest-price 分配策略启动竞价型实例

以下 create-auto-scaling-group 命令会创建指定以下内容的 Auto Scaling 组：

- 作为按需实例启动的组的百分比 (50)。（这不会指定要开始使用的按需型实例的基本数量。）
- 按优先级顺序（c5.large、c5a.large、m5.large、m5a.large、c4.large、m4.large、c3.large、m3.large）启动的实例类型。
- 要在其中启动实例的子网（subnet-5ea0c127、subnet-6194ea3b、subnet-c934b782）。每个子网都对应不同的可用区。
- 启动模板 (my-launch-template) 和启动模板版本 ($Latest)。

Amazon EC2 Auto Scaling 尝试满足您的按需容量时，它会首先启动 c5.large 实例类型。对于您的竞价容量，Amazon EC2 Auto Scaling 尝试在每个可用区中的两个价格最低的池中平均启动竞价型实例。

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

下面是一个 config.json 示例文件。

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 50,
            "SpotAllocationStrategy": "lowest-price",
            "SpotInstancePools": 2
        }
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 3,
```

```
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

或者，您可以使用以下 create-auto-scaling-group 命令创建自动扩缩组。这将引用 YAML 文件作为自动扩缩组的唯一参数，而不是 JSON 文件。

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

下面是一个 config.yaml 示例文件。

```
---
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
  InstancesDistribution:
    OnDemandPercentageAboveBaseCapacity: 50
    SpotAllocationStrategy: lowest-price
    SpotInstancePools: 2
MinSize: 1
MaxSize: 5
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

示例 4：使用 price-capacity-optimized 分配策略启动 Spot 实例

以下 create-auto-scaling-group 命令会创建指定以下内容的 Auto Scaling 组：

- 作为按需实例启动的组的百分比 (30)。（这不会指定要开始使用的按需型实例的基本数量。）
- 按优先级顺序
  （c5.large、c5a.large、m5.large、m5a.large、c4.large、m4.large、c3.large、m3.large）
  启动的实例类型。
- 要在其中启动实例的子网（subnet-5ea0c127、subnet-6194ea3b、subnet-c934b782）。每个子网都对应不同的可用区。
- 启动模板 (my-launch-template) 和启动模板版本 ($Latest)。

Amazon EC2 Auto Scaling 尝试满足您的按需容量时，它会首先启动 c5.large 实例类型。对于您的竞价型容量，Amazon EC2 Auto Scaling 将尝试从竞价型实例池启动竞价型实例，这些竞价型实例池不仅具有尽可能低的价格，对于正在启动的实例数量而言还具有最佳容量

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

下面是一个 config.json 示例文件。

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 30,
            "SpotAllocationStrategy": "price-capacity-optimized"
        }
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 3,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

或者，您可以使用以下 create-auto-scaling-group 命令创建自动扩缩组。这将引用 YAML 文件作为自动扩缩组的唯一参数，而不是 JSON 文件。

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

下面是一个 config.yaml 示例文件。

```
---
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
```

```
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
  InstancesDistribution:
    OnDemandPercentageAboveBaseCapacity: 30
    SpotAllocationStrategy: price-capacity-optimized
MinSize: 1
MaxSize: 5
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Note

有关其他示例，请参阅 为实例类型指定不同的启动模板 (p. 64)、为 Amazon EC2 Auto Scaling 配置实例权重 (p. 67) 和 使用容量重新平衡来处理 Amazon EC2 竞价型实例中断 (p. 262)。

验证您的自动扩缩组配置和已启动的实例 (Amazon CLI)

要检查 Auto Scaling 组是否已正确配置以及它是否已启动实例，请使用 describe-auto-scaling-groups 命令。验证混合实例策略和子网列表是否存在且配置正确。如果实例已启动，您将看到实例及其状态的列表。要查看实例启动所产生的扩展活动，请使用 describe-scaling-activities 命令。您可以监控正在进行和最近完成的扩展活动。有关更多信息，请参阅验证 Auto Scaling 组的扩缩活动 (p. 160)。

配置覆盖

覆盖是您对启动模板属性所做的更改。Amazon EC2 Auto Scaling 支持覆盖实例类型属性。这样，您就可以指定多种实例类型。您还可以定义多个启动模板，以允许具有不同 CPU 架构（例如 Arm 和 x86）的实例在同一自动扩缩组中启动。为此，您必须将 `Overrides` 结构添加到混合实例策略。

`Overrides` 结构允许您定义一组 Amazon EC2 Auto Scaling 可用于启动实例的参数，包括：

- `InstanceType` — 实例类型。有关可用实例类型的更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的实例类型。
- `LaunchTemplateSpecification` —（可选）单个实例类型的启动模板。此属性当前限制为 Amazon CLI 或 SDK，并且不可从控制台使用。
- `WeightedCapacity` —（可选）此类型的预置实例为满足 Auto Scaling 组的所需容量而提供的单位数量。如果您为一种实例类型指定权重值，则必须为所有实例类型指定权重值。

Note

作为指定实例类型列表的替代方法，您可以指定一组实例属性，用作选择 Auto Scaling 组使用的实例类型的标准。这称为基于属性的实例类型选择。有关更多信息，请参阅使用基于属性的实例类型选择 (p. 72)。

目录

- 为实例类型指定不同的启动模板 (p. 64)
- 为 Amazon EC2 Auto Scaling 配置实例权重 (p. 67)

为实例类型指定不同的启动模板

Overrides 结构允许您为新的或现有 Auto Scaling 组定义单个实例类型的新启动模板。例如，如果实例类型的架构需要与该组其余部分不同的 AMI，则必须指定具有兼容 AMI 的启动模板。

假设您为计算密集型应用程序配置 Auto Scaling 组，并希望混合 C5、C5a 和 C6g 实例类型。但是，C6g 实例具有基于 64 位 Arm 架构的 Amazon 引力处理器，而 C5 和 C5a 实例在 64 位 Intel x86 处理器上运行。C5 实例的 AMI 适用于 C5a 实例，反之亦然，但不适用于 C6g 实例。Overrides 属性允许您为 C6g 实例包含不同的启动模板，同时对 C5 和 C5a 实例仍使用相同的启动模板。

> **Note**
>
> 目前，仅当您使用 Amazon CLI 或 SDK 时此功能才可用，并且不可从控制台使用。

目录

- 添加或更改实例类型的启动模板（Amazon CLI）(p. 64)

添加或更改实例类型的启动模板（Amazon CLI）

以下过程显示如何使用 Amazon CLI 配置 Auto Scaling 组，以便一个或多个实例类型使用与该组其余部分不同的启动模板。

创建和配置新的 Auto Scaling 组

1. 创建一个配置文件，您可以在其中指定混合实例策略结构并包含 Overrides 结构。

   以下是采用 JSON 格式的示例配置文件的内容。它指定 c5.large、c5a.large 和 c6g.large 实例类型，并为 c6g.large 实例类型定义新的启动模板，以确保使用适当的 AMI 启动 Arm 实例。Amazon EC2 Auto Scaling 使用实例类型顺序确定在满足按需容量时应首先使用哪一实例类型。

```
{
  "AutoScalingGroupName":"my-asg",
  "MixedInstancesPolicy":{
    "LaunchTemplate":{
      "LaunchTemplateSpecification":{
        "LaunchTemplateName":"my-launch-template-for-x86",
        "Version":"$Latest"
      },
      "Overrides":[
        {
          "InstanceType":"c6g.large",
          "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template-for-arm",
            "Version": "$Latest"
          }
        },
        {
          "InstanceType":"c5.large"
        },
        {
          "InstanceType":"c5a.large"
        }
      ]
    },
    "InstancesDistribution":{
      "OnDemandBaseCapacity": 1,
      "OnDemandPercentageAboveBaseCapacity": 50,
      "SpotAllocationStrategy": "capacity-optimized"
    }
  },
  "MinSize":1,
```

```
  "MaxSize":5,
  "DesiredCapacity":3,
  "VPCZoneIdentifier":"subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
  "Tags":[ ]
}
```

2. 使用以下 create-auto-scaling-group 命令，引用 JSON 文件作为 Auto Scaling 组的唯一参数。

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

更改现有 Auto Scaling 组中实例类型的启动模板

- 通过传递 `Overrides` 结构，使用以下 update-auto-scaling-group 命令为实例类型指定不同的启动模板。

  进行此更改后，启动的任何新实例将基于新设置，但现有实例不受影响。为确保 Auto Scaling 组正在使用新设置，您可以通过启动实例刷新或使用最大实例生命周期功能来替换组中的所有实例。

```
aws autoscaling update-auto-scaling-group --cli-input-json file://~/config.json
```

  下面是一个 `config.json` 示例文件。

```
{
  "AutoScalingGroupName":"my-asg",
  "MixedInstancesPolicy":{
    "LaunchTemplate":{
      "Overrides":[
        {
          "InstanceType":"c6g.large",
          "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template-for-arm",
            "Version": "$Latest"
          }
        },
        {
          "InstanceType":"c5.large"
        },
        {
          "InstanceType":"c5a.large"
        }
      ]
    }
  }
}
```

验证 Auto Scaling 组的启动模板

- 使用以下 describe-auto-scaling-groups 命令验证和查看当前指定的启动模板。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

  以下为响应示例。

```
{
  "AutoScalingGroups":[
    {
      "AutoScalingGroupName":"my-asg",
```

```
"AutoScalingGroupARN":"arn",
"MixedInstancesPolicy":{
  "LaunchTemplate":{
    "LaunchTemplateSpecification":{
      "LaunchTemplateId":"lt-0fb0e487336917fb2",
      "LaunchTemplateName":"my-launch-template-for-x86",
      "Version":"$Latest"
    },
    "Overrides":[
      {
        "InstanceType":"c6g.large",
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-09d958b8fb2ba5bcc",
            "LaunchTemplateName": "my-launch-template-for-arm",
            "Version": "$Latest"
        }
      },
      {
        "InstanceType":"c5.large"
      },
      {
        "InstanceType":"c5a.large"
      }
    ]
  },
  "InstancesDistribution":{
    "OnDemandAllocationStrategy":"prioritized",
    "OnDemandBaseCapacity":1,
    "OnDemandPercentageAboveBaseCapacity":50,
    "SpotAllocationStrategy":"capacity-optimized"
  }
},
"MinSize":1,
"MaxSize":5,
"DesiredCapacity":3,
"Instances":[
  {
    "InstanceId":"i-07c63168522c0f620",
    "InstanceType":"c5.large",
    "AvailabilityZone":"us-west-2c",
    "LifecycleState":"InService",
    "HealthStatus":"Healthy",
    "LaunchTemplate":{
      "LaunchTemplateId":"lt-0fb0e487336917fb2",
      "LaunchTemplateName":"my-launch-template-for-x86",
      "Version":"1"
    },
    "ProtectedFromScaleIn":false
  },
  {
    "InstanceId":"i-0b7ff78be9896a2c2",
    "InstanceType":"c5.large",
    "AvailabilityZone":"us-west-2a",
    "LifecycleState":"InService",
    "HealthStatus":"Healthy",
    "LaunchTemplate":{
      "LaunchTemplateId":"lt-0fb0e487336917fb2",
      "LaunchTemplateName":"my-launch-template-for-x86",
      "Version":"1"
    },
    "ProtectedFromScaleIn":false
  },
  {
    "InstanceId":"i-0c682c2ceae918bc0",
    "InstanceType":"c6g.large",
    "AvailabilityZone":"us-west-2b",
```

```
          "LifecycleState":"InService",
          "HealthStatus":"Healthy",
          "LaunchTemplate":{
            "LaunchTemplateId":"lt-09d958b8fb2ba5bcc",
            "LaunchTemplateName":"my-launch-template-for-arm",
            "Version":"1"
          },
        ...
    }
  ]
}
```

为 Amazon EC2 Auto Scaling 配置实例权重

在将 Auto Scaling 组配置为启动多个实例类型时，您可以选择使用实例权重定义每个实例对所需的组容量贡献的容量单位数。这样，您可以通过直接映射到应用程序性能的方式指定每种实例类型的相对权重。您可以对实例加权以满足您的特定应用程序需求，例如，依据内核 (vCPU) 或内存 (GiB) 加权。

例如，假设您运行一个计算密集型应用程序，该应用程序在至少具有 8 个 vCPU 和 15 GiB RAM 时性能最佳。如果您使用 `c5.2xlarge` 作为基本单元，以下任何 EC2 实例类型都将满足您的应用程序需求。

实例类型示例

实例类型	vCPU	内存（GiB）
`c5.2xlarge`	8	16
`c5.4xlarge`	16	32
`c5.12xlarge`	48	96
`c5.18xlarge`	72	144
`c5.24xlarge`	96	192

默认情况下，所有实例类型都被视为具有相同权重。换句话说，无论 Amazon EC2 Auto Scaling 启动大型还是小型实例类型，每个实例都会计入组的所需容量。

但是，通过实例权重，您可以分配一个数值，该值指定与每个实例类型关联的容量单位数。例如，如果实例大小不相同，`c5.2xlarge` 实例的权重可能为 2，而 `c5.4xlarge`（大两倍）的权重可能为 4，依此类推。然后，在 Amazon EC2 Auto Scaling 启动实例时，这些实例的权重会计入您所需的容量中。

目录

- 每单位小时价格 (p. 67)
- 注意事项 (p. 68)
- 添加或修改 Auto Scaling 组的权重 (p. 69)
- 其他信息 (p. 72)

每单位小时价格

下表比较美国东部（弗吉尼亚北部、俄亥俄）不同可用区中竞价型实例的每小时价格和同一区域中按需实例的价格。显示的价格是示例定价，而不是当前定价。这些价格是您每实例小时的成本。

示例：每实例小时的竞价定价

实例类型	us–east–1a	us–east–1b	us–east–1c	按需定价
c5.2xlarge	0.180 USD	0.191 USD	0.170 USD	0.34 USD
c5.4xlarge	0.341 USD	0.361 USD	0.318 USD	0.68 USD
c5.12xlarge	0.779 USD	0.777 USD	0.777 USD	2.04 USD
c5.18xlarge	1.207 USD	1.475 USD	1.357 USD	3.06 USD
c5.24xlarge	1.555 USD	1.555 USD	1.555 USD	4.08 USD

通过实例权重，您可以根据每单位小时 的使用量来评估您的成本。您可以将某种实例类型的价格除以它表示的单位数来确定每单位小时价格。对于按需实例，部署一个实例类型时的每单位小时 价格与部署不同大小的相同实例类型时的价格相同。但是，相比之下，每单位小时 的 Spot 价格因竞价池而异。

了解每单位小时 价格计算如何与权重实例配合工作的最简单方法是使用示例。例如，为便于计算，假定您只想启动 us-east-1a 中的竞价型实例。每单位小时价格如下所示。

示例：每单位小时 Spot 价格示例

实例类型	us–east–1a	实例权重	每单位小时价格
c5.2xlarge	0.180 USD	2	0.090 USD
c5.4xlarge	0.341 USD	4	0.085 USD
c5.12xlarge	0.779 USD	12	0.065 USD
c5.18xlarge	1.207 USD	18	0.067 USD
c5.24xlarge	1.555 USD	24	0.065 USD

注意事项

本节讨论有效实施实例权重的关键注意事项。

- 首先选择能够反映应用程序实际性能要求的几种实例类型。然后，通过指定实例类型的权重，确定每种实例类型应计入 Auto Scaling 组所需容量的数量。该权重应用于组中的当前和未来实例。
- 在为权重选择超大范围时应小心谨慎。例如，当较大实例类型的权重为 200 时，我们建议不要将前一个实例类型的权重指定为 1。此外，最小权重和最大权重之间的差异不能太极端。如果任何实例类型的权重差异太大，这可能会对持续的性价比优化产生负面影响。
- Auto Scaling 组大小以容量单位计量，而不是以实例计量。例如，如果权重基于 vCPU，则必须指定所需核心数、最小核心数和最大核心数。
- 设置权重和所需容量，使所需容量至少比最大权重大两到三倍。

通过实例权重，引入了以下新行为：

- 当前容量等于或高于所需容量。由于 Amazon EC2 Auto Scaling 希望预配置实例直到完全达到所需容量，因此可能会超过目标。例如，假设您指定了两个实例类型 c5.2xlarge 和 c5.12xlarge，并且您为 c5.2xlarge 分配了实例权重 2，为 c5.12xlarge 分配了实例权重 12。如果还剩 5 个单位才能实现所需容量，Amazon EC2 Auto Scaling 会预配置 c5.12xlarge，所需容量将超过 7 个单位。
- 当 Amazon EC2 Auto Scaling 预配置实例以达到所需容量时，同避免超过目标相比，跨可用区分配实例和遵从按需实例与竞价型实例的分配策略都会具有更高优先级。

- 通过使用您的首选分配策略，Amazon EC2 Auto Scaling 可以超出最大容量限制，以便在可用区之间保持平衡。Amazon EC2 Auto Scaling 强制执行的硬性限制是一个值，该值等于所需容量加上最大重量的和。

添加或修改现有组的权重时，请注意以下事项：

- 向现有 Auto Scaling 组添加实例权重时，您必须包含已在组中运行的任何实例类型。
- 修改现有实例权重时，Amazon EC2 Auto Scaling 将根据新权重启动或终止实例以达到所需容量。
- 如果您删除了实例类型，该实例类型的所有正在运行的实例都将继续采用其上次更新的权重值，即使该实例类型已被删除也是如此。

添加或修改 Auto Scaling 组的权重

您可以向现有 Auto Scaling 组添加权重，也可以在创建新的 Auto Scaling 组时添加权重。您还可以更新现有 Auto Scaling 组以定义新的配置选项（竞价/按需使用情况、竞价分配策略、实例类型）。如果您更改了所需的竞价型或按需型实例数量，Amazon EC2 Auto Scaling 会逐步替换现有实例以匹配新的购买选项。

在使用实例权重创建 Auto Scaling 组之前，我们建议您熟悉启动具有多种实例类型的组。有关更多信息和其他示例，请参阅 Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。

以下示例说明如何使用 Amazon CLI 在您创建 Auto Scaling 组时添加权重，以及添加或修改现有 Auto Scaling 组的权重。您可以在 JSON 文件中配置各种参数，然后将 JSON 文件作为 Auto Scaling 组的唯一参数引用。

在创建时向 Auto Scaling 组添加权重

- 使用 create-auto-scaling-group 命令创建新的 Auto Scaling 组。例如，以下命令创建一个新的 Auto Scaling 组，并通过指定以下内容添加实例权重：
  - 作为按需实例启动的组的百分比 (0)
  - 每个可用区中竞价型实例的分配策略(capacity-optimized)。
  - 按优先级顺序启动（m4.16xlarge、m5.24xlarge）的实例类型
  - 与实例类型之间的相对大小差异 (vCPU) 相对应的实例权重（16、24）
  - 要在其中启动实例的子网（subnet-5ea0c127、subnet-6194ea3b、subnet-c934b782），每个子网对应于不同的可用区
  - 启动模板 (my-launch-template) 和启动模板版本 ($Latest)

```
aws autoscaling create-auto-scaling-group --cli-input-json file:///~/config.json
```

下面是一个 config.json 示例文件。

```
{
    "AutoScalingGroupName": "my-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "m4.16xlarge",
                    "WeightedCapacity": "16"
                },
                {
                    "InstanceType": "m5.24xlarge",
```

```
                    "WeightedCapacity": "24"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 0,
            "SpotAllocationStrategy": "capacity-optimized"
        }
    },
    "MinSize": 160,
    "MaxSize": 720,
    "DesiredCapacity": 480,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "Tags": []
}
```

添加或修改现有 Auto Scaling 组的权重

- 使用 update-auto-scaling-group 命令添加或修改权重。例如，以下命令通过指定以下内容向现有 Auto Scaling 组中的实例类型添加权重：
  - 按优先级顺序启动（c5.18xlarge、c5.24xlarge、c5.2xlarge、c5.4xlarge）的实例类型
  - 与实例类型之间的相对大小差异 (vCPU) 相对应的实例权重（18、24、2、4）
  - 新增加的所需容量，大于最大权重

```
aws autoscaling update-auto-scaling-group --cli-input-json file://~/config.json
```

下面是一个 config.json 示例文件。

```
{
    "AutoScalingGroupName": "my-existing-asg",
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "Overrides": [
                {
                    "InstanceType": "c5.18xlarge",
                    "WeightedCapacity": "18"
                },
                {
                    "InstanceType": "c5.24xlarge",
                    "WeightedCapacity": "24"
                },
                {
                    "InstanceType": "c5.2xlarge",
                    "WeightedCapacity": "2"
                },
                {
                    "InstanceType": "c5.4xlarge",
                    "WeightedCapacity": "4"
                }
            ]
        }
    },
    "MinSize": 0,
    "MaxSize": 100,
    "DesiredCapacity": 100
}
```

验证 Auto Scaling 组的权重

- 使用以下 describe-auto-scaling-groups 命令验证权重。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

以下为响应示例。

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            "MixedInstancesPolicy": {
                "LaunchTemplate": {
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "$Latest"
                    },
                    "Overrides": [
                        {
                            "InstanceType": "m4.16xlarge",
                            "WeightedCapacity": "16"
                        },
                        {
                            "InstanceType": "m5.24xlarge",
                            "WeightedCapacity": "24"
                        }
                    ]
                },
                "InstancesDistribution": {
                    "OnDemandAllocationStrategy": "prioritized",
                    "OnDemandBaseCapacity": 0,
                    "OnDemandPercentageAboveBaseCapacity": 0,
                    "SpotAllocationStrategy": "capacity-optimized"
                }
            },
            "MinSize": 160,
            "MaxSize": 720,
            "DesiredCapacity": 480,
            "DefaultCooldown": 300,
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2b",
                "us-west-2c"
            ],
            "LoadBalancerNames": [],
            "TargetGroupARNs": [],
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": 0,
            "Instances": [
                {
                    "InstanceId": "i-027327f0ace86f499",
                    "InstanceType": "m5.24xlarge",
                    "AvailabilityZone": "us-west-2a",
                    "LifecycleState": "InService",
                    "HealthStatus": "Healthy",
                    "LaunchTemplate": {
                        "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "7"
                    },
```

```
                    "ProtectedFromScaleIn": false,
                    "WeightedCapacity": "24"
                },
                {
                    "InstanceId": "i-0ec0d761cc134878d",
                    "InstanceType": "m4.16xlarge",
                    "AvailabilityZone": "us-west-2a",
                    "LifecycleState": "Pending",
                    "HealthStatus": "Healthy",
                    "LaunchTemplate": {
                        "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "7"
                    },
                    "ProtectedFromScaleIn": false,
                    "WeightedCapacity": "16"
                },
            ...
        }
    ]
}
```


其他信息

实例权重和分配策略

分配策略 (p. 50)确定您的实例来自哪个实例池。在使用实例权重功能时，分配策略的作用与用于其他 Auto Scaling 组时完全相同。但是，在使用 `lowest-price` 策略时，如何选择实例池存在一个关键区别。为您的分配策略选择 `lowest-price` 时，您的实例来自每个可用区中单位价格最低的实例池。

例如，假设您有一个 Auto Scaling 组，该组具有多种实例类型，每个实例类型带有不同数量的 vCPU。针对竞价和按需分配策略使用 `lowest-price`。如果您选择根据每种实例类型的 vCPU 计数分配权重，Amazon EC2 Auto Scaling 将启动交付时每个分配权重值（例如，每个 vCPU）价格最低的任何实例类型。如果是 Spot 实例，那么这意味着每个 vCPU 的最低 Spot 价格。如果是按需型实例，那么这意味着每个 vCPU 的最低按需价格。

实例权重和竞价最高价

在使用 Amazon CLI 或 SDK 创建 Auto Scaling 组时，您可以指定 `SpotMaxPrice` 参数。此参数决定了您愿意为竞价型实例小时支付的最高价，默认情况下，这将设置为按需价格。当您使用实例权重功能时，请记住，最高 Spot 价格反映的是最高单价（例如，每个 vCPU 的价格），而不是整个实例的最高价格。

示例：使用基于属性的实例类型选择创建 Auto Scaling 组

在创建自动扩缩组时，必须指定信息以配置以下内容：

- 为 Amazon EC2 实例指定 AMI 和实例类型的启动模板
- 实例的可用区和 VPC 子网
- 所需容量
- 最小容量和最大容量限制

作为在创建混合实例组 (p. 50)时手动选择实例类型的替代方法，您可以指定一组用于描述计算要求的实例属性。当 Amazon EC2 Auto Scaling 启动实例时，Auto Scaling 组使用的所有实例类型都必须与您所需实例的属性匹配。这称为基于属性的实例类型选择。

您的自动扩缩组或启动模板指定了您的实例属性。这些属性包括您打算在实例上运行的应用程序所需的存储容量和计算能力。此外，您的自动扩缩组或启动模板为您可以自定义的竞价型实例和按需型实例指定了两个价格保护阈值。通过这种方式，您可以在不需要时防止 Amazon EC2 Auto Scaling 启动更昂贵的实例类型。

这种方法非常适合于可以灵活处理所使用实例类型的工作负载和框架，例如容器、大数据和 CI/CD。

选择基于属性的实例类型具有以下优势：

- Amazon EC2 Auto Scaling 可以从各种实例类型中进行选择，从而启动竞价型实例。这是符合灵活处理实例类型的竞价型实例的最佳实践，可以使得 Amazon EC2 竞价型服务有更好的机会查找和分配所需的计算容量。
- 因为可供选择的选项如此之多，为您的工作负载找到合适的实例类型可能会非常耗时。通过指定实例属性，您可以在配置混合实例组时简化实例类型的选择。
- 您的 Auto Scaling 组可以在发布新一代实例类型时就对其进行采用。新一代实例类型将自动投入使用，如果它们符合您的要求并与您为 Auto Scaling 组选择的分配策略保持一致。

您可以通过 Amazon Web Services Management Console、Amazon CLI 或 SDK 使用基于属性的实例类型选择。

有关如何在启动模板中配置基于属性的实例类型选择的信息，请参阅 为 Auto Scaling 组创建启动模板 (p. 20)。您还可以通过开发工具包在 Amazon EC2 Auto Scaling API 调用中传递参数，以配置基于属性的实例类型选择。有关更多信息，请参阅《Amazon EC2 Auto Scaling API 参考》中的 InstanceRequirements。

要详细了解基于属性的实例类型选择，请参阅 Amazon 博客中的适用于 EC2 Auto Scaling 和 EC2 机群的基于属性的实例类型选择。

目录

- 注意事项 (p. 73)
- 先决条件 (p. 74)
- 使用基于属性的实例类型选择 (p. 74)
- 限制 (p. 78)

注意事项

使用基于属性的实例类型选择时需要考虑以下几点：

- 对于大多数通用型工作负载，指定您所需要的 vCPU 和内存数量就足够了。对于高级使用案例，您可以指定存储类型、网络接口、CPU 制造商和加速器类型等属性。
- 默认情况下，您将 Auto Scaling 组的所需容量值设置为实例的数量。在使用基于属性的实例类型选择时，您可以选择将所需容量类型指定为 vCPU 的数量或内存量。然后，在 Amazon EC2 Auto Scaling 启动实例时，它们的 vCPU 数量或内存量会计入您所需的容量中。在 Amazon EC2 Auto Scaling 控制台中创建您的组时，此设置将显示在 Configure group size and scaling policies（配置组大小和扩展策略）页面的 Group size（组大小）部分中。此功能是实例权重 (p. 67)功能的有用替代。
- 您可以预览符合计算要求的实例类型而无需启动它们，并在必要时调整要求。在 Amazon EC2 Auto Scaling 控制台中创建 Auto Scaling 组时，实例类型的预览将显示在 Choose instance launch options（选择实例启动选项）页面的 Preview matching instance types（预览匹配的实例类型）部分中。
- 或者，您可以通过使用 Amazon CLI 或者 SDK 建立 Amazon EC2 GetInstanceTypesFromInstanceRequirements API 调用来预览实例类型。在请求中传递 `InstanceRequirements` 参数，其格式与用于创建或更新 Auto Scaling 组的格式一致。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的预览具有指定属性的实例类型。

了解价格保护

价格保护是一项功能，可以保护您的 Auto Scaling 组免受实例类型之间的极端价格差异影响。当您使用基于属性的实例类型选择创建新的 Auto Scaling 组或更新现有 Auto Scaling 组时，我们将默认启用价格保护。或者，您可以为竞价型和按需型实例选择价格保护阈值。当您执行此操作时，Amazon EC2 Auto Scaling 不会选择价格高于您指定阈值的实例类型。阈值代表您愿意支付的金额，其定义是高于基准的百分比，而不是绝对值。基准取决于具有指定属性的最便宜当前一代 M、C 或 R 实例类型的价格。如果您的属性与任何 M、C 或 R 实例类型都不匹配，我们将使用价格最低的实例类型。

如果您没有指定阈值，则默认使用以下阈值：

- 对于按需型实例，价格保护阈值设置为 20%。
- 对于竞价型实例，价格保护阈值设置为 100%。

在 Amazon EC2 Auto Scaling 控制台中创建自动扩缩组时，您可以更新这些值。在 Choose instance launch options（选择实例启动选项）页面上，从 Additional instance attributes（其他实例属性）下拉列表中选择所需的价格保护属性。然后，在文本框中键入或选择该属性的值。您也可以稍后通过从控制台编辑自动扩缩组或使用 Amazon CLI 或开发工具包更新这些值。

> **Note**
>
> 如果您将 Desired capacity type（所需容量类型）设置为 vCPUs (vCPU) 或 Memory GiB（内存 GiB），则将根据单位 vCPU 价格或单位内存价格而不是单位实例价格来应用价格保护阈值。

先决条件

创建启动模板，其中包括启动 EC2 实例所需的参数，例如 Amazon Machine Image (AMI) 和安全组。有关更多信息，请参阅为 Auto Scaling 组创建启动模板 (p. 20)。

验证启动模板尚未请求竞价型实例。

验证您拥有使用启动模板所需的权限。当您使用启动模板时，将检查您的 `ec2:RunInstances` 权限。如果启动模板指定了 IAM 角色，也会检查您的 `iam:PassRole` 权限。有关更多信息，请参阅 启动模板支持 (p. 328)。

使用基于属性的实例类型选择

完成以下步骤以创建您可以扩展的竞价型实例和按需型实例队组。

通过以下程序使用基于属性的实例类型选择。如果您更愿意选择您的组可以启动的单个实例类型，则请参阅 Auto Scaling 组具有多个实例类型和购买选项 (p. 50) 以通过手动选择实例类型来配置实例类型要求。

以下步骤说明了如何创建使用基于属性的实例类型选择的自动扩缩组。

- 指定用于启动实例的启动模板。
- 选择要在其中启动您的自动扩缩组的 VPC 和子网。
- 选择使用新要求覆盖启动模板的现有实例类型要求的选项。
- 指定符合您的计算需求的实例属性，例如 vCPU 数、内存大小和存储容量。
- 指定要启动的按需型实例和竞价型实例的百分比。
- 选择决定 Amazon EC2 Auto Scaling 将如何利用可能的实例类型来达到您的按需型和竞价型容量的分配策略。
- 指定组大小，包括您的所需容量、最小容量、最大容量以及所需容量类型，这定义了所需容量的测量单位。

使用基于属性的实例类型选择（控制台）创建 Auto Scaling 组

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 在屏幕顶部的导航栏中，选择在创建启动模板时使用的同一 Amazon Web Services 区域。
3. 选择 Create an Auto Scaling group (创建 Auto Scaling 组)。
4. 在选择启动模板或配置页面上，对于 Auto Scaling 组名称，输入 Auto Scaling 组的名称。
5. 要选择启动模板，请执行以下操作：
   a. 对于启动模板，请选择现有启动模板。
   b. 对于 Launch template version（启动模板版本），选择 Auto Scaling 组在扩展时使用启动模板的默认版本、最新版本还是特定版本。
   c. 验证您的启动模板是否支持您计划使用的所有选项，然后选择下一步。
6. 要选择要在其中启动实例的 VPC 和子网，请执行以下操作：
   a. 在 Choose instance launch options (选择实例启动选项) 页面的 Network (网络) 下方，对于 VPC，选择相应的 VPC。 Auto Scaling 组必须与您在启动模板中指定的安全组创建在相同的 VPC 中。
   b. 对于 Availability Zones and subnets (可用区和子网)，选择指定 VPC 中的一个或多个子网。可以在多个可用区中使用子网以提供高可用性。有关更多信息，请参阅选择 VPC 子网时的注意事项 (p. 302)。
7. 要将您的组配置为使用基于属性的实例类型选择，请执行以下操作：
   a. 对于 Instance type requirements（实例类型要求），请选择 Override launch template（覆盖启动模板）。

      Note

      如果您选择了某个已经包含您的要求的启动模板，则启动模板设置（例如 vCPU 数和内存大小）将被自动用作属性。您可以随时通过 Amazon EC2 Auto Scaling 控制台更新这些属性。

   b. 在 Specify instance attributes（指定实例属性）下，首先输入 vCPU 和内存要求。
      - vCPU：输入计算要求的最小 vCPU 数量和最大 vCPU 数量。选择 No minimum（无最小值）或 No maximum（无最大值）复选框以指示该方向没有限制。
      - Memory (MiB)（内存 (MiB)）：为您的计算要求输入最小和最大存储容量（以 MiB 为单位）。选择 No minimum（无最小值）或 No maximum（无最大值）复选框以指示该方向没有限制。
   c. （可选）对于 Additional instance attributes（额外实例属性），选择 Add Attributes（添加属性）以更详细地表达您的计算要求。您在此处选择的属性和值进一步定义了可以启动的实例类型。

      有关所有受支持属性的更多信息，请参阅《Amazon EC2 Auto Scaling API 参考》中的 InstanceRequirements。
   d. 对于 Preview matching instance types（预览匹配的实例类型），请查看符合指定计算要求的实例类型，例如 vCPU、内存和存储。
   e. 在 Instance purchase options（实例购买选项）下，对于 Instances distribution（实例分配），分别指定要启动的按需型实例和竞价型实例在该组中所占的百分比。如果您的应用程序无状态、容错，并且可以处理中断的实例，则可以指定更高的竞价型实例百分比。
   f. 如果您指定了一个竞价型实例百分比，则可以选中 Include On-Demand base capacity（包括按需基本容量）旁边的复选框，然后指定按需型实型例在自动扩缩组的最小初始容量中必须达到的数量。超出基本容量的任何内容都使用百分比来确定要启动多少按需实例和 Spot 实例。
   g. 在 Allocation strategies（分配策略）下，已为 On-Demand allocation strategy（按需分配策略）自动选择 Lowest price（最低价格），无法对其进行更改。
   h. 对于 Spot allocation strategy（竞价型分配策略），选择分配策略。默认情况下，将选择 Price capacity optimized（价格容量优化）。默认情况下，将隐藏 Lowest price（最低价格），仅在您选择 Show all strategies（显示所有策略）时才会出现。

Note

如果您选择了 Lowest price（最低价格），则请输入价格最低的池的数量，以跨价格最低的池实现多样化。

i. 对于 Capacity rebalance (容量再平衡)，选择启用还是禁用容量再平衡。

如果您选择了一个竞价型实例百分比，则可以使用容量再平衡功能，以自动响应竞价型实例因竞价中断而即将终止的情况。有关更多信息，请参阅使用容量重新平衡来处理 Amazon EC2 竞价型实例中断 (p. 262)。

j. 完成之后，选择 Next（下一步）两次，以转至 Configure group size and scaling policies（配置组大小和扩缩策略）页面。

8. 对于 Configure group size and scaling policies（配置组大小和扩缩策略）步骤，请执行以下操作：

a. 如果您希望以实例以外的其他单位来衡量所需容量，请为 Desired capacity type（所需容量类型）选择相应的选项。支持 Units（单位）、vCPUs（vCPU 数）和 Memory GiB（内存 GiB）。默认情况下，Amazon EC2 Auto Scaling 指会定 Units（单位），这会用实例数量来衡量。

b. 为 Desired capacity（所需容量）输入您的 Auto Scaling 组的初始大小，然后根据需要更新 Minimum capacity（最小容量）和 Maximum capacity（最大容量）。有关更多信息，请参阅为 Auto Scaling 组设置容量限制 (p. 122)。

c. （可选）将组配置为通过指定目标跟踪扩展策略进行扩展。或者，在您创建组后指定此策略。有关更多信息，请参阅Amazon EC2 Auto Scaling 的目标跟踪扩缩策略 (p. 133)。

d. （可选）启用实例横向缩减保护功能，该功能可以防止您的自动扩缩组在横向缩减时终止实例。有关更多信息，请参阅实例横向缩减保护 (p. 230)。

e. 在完成后，选择下一步。

9. （可选）要在自动扩缩组扩展时收到通知，请为 Add notification（添加通知）配置通知，然后选择 Next（下一步）。有关更多信息，请参阅在 Auto Scaling 组扩展时获取 Amazon SNS 通知 (p. 258)。

10. （可选）要添加标签，请选择 Add tag（添加标签），为每个标签提供标签键和值，然后选择 Next（下一步）。有关更多信息，请参阅为 Auto Scaling 组和实例添加标签 (p. 87)。

11. 在 Review（查看）页面上，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

示例：使用基于属性的实例类型选择创建 Auto Scaling 组 (Amazon CLI)

要使用命令行创建具有基于属性的实例类型选择的 Auto Scaling 组，您可以使用以下 create-auto-scaling-group 命令。

指定了以下实例属性：

- `VCpuCount` – 实例类型的 vCPU 数量必须最少为 4 个，最多为 8 个。
- `MemoryMiB` – 实例类型的最小内存必须为 16384 MiB。
- `CpuManufacturers` - 实例类型必须具有英特尔制造的 CPU。

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

下面是一个 `config.json` 示例文件。

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredCapacityType": "units",
```

```
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Default"
            },
            "Overrides": [{
                "InstanceRequirements": {
                    "VCpuCount": {"Min": 4, "Max": 8},
                    "MemoryMiB": {"Min": 16384},
                    "CpuManufacturers": ["intel"]
                }
            }]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 50,
            "SpotAllocationStrategy": "price-capacity-optimized"
        }
    },
    "MinSize": 0,
    "MaxSize": 100,
    "DesiredCapacity": 4,
    "DesiredCapacityType": "units",
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

要将所需容量的值设置为 vCPU 的数量或内存量，请在文件中指定 "DesiredCapacityType": "vcpu" 或 "DesiredCapacityType": "memory-mib"。默认的所需容量类型为 units，它将所需容量的值设置为实例数。

YAML

或者，您可以使用以下 create-auto-scaling-group 命令创建自动扩缩组。这将引用 YAML 文件作为自动扩缩组的唯一参数，而不是 JSON 文件。

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

下面是一个 config.yaml 示例文件。

```
---
AutoScalingGroupName: my-asg
DesiredCapacityType: units
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceRequirements:
        VCpuCount:
          Min: 2
          Max: 4
        MemoryMiB:
          Min: 2048
        CpuManufacturers:
        - intel
  InstancesDistribution:
    OnDemandPercentageAboveBaseCapacity: 50
    SpotAllocationStrategy: price-capacity-optimized
MinSize: 0
MaxSize: 100
DesiredCapacity: 4
DesiredCapacityType: units
```

```
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

要将所需容量的值设置为 vCPU 的数量或内存量，请在文件中指定 `DesiredCapacityType: vcpu` 或 `DesiredCapacityType: memory-mib`。默认的所需容量类型为 `units`，它将所需容量的值设置为实例数。

限制

- 您只能为使用启动模板的 Auto Scaling 组配置基于属性的实例类型选择。
- 如果您现有的 Auto Scaling 组并计划用所需的实例属性替换实例类型，则按需分配策略必须为 `lowest-price`。要使用 `prioritized` 分配策略，您必须继续手动添加实例类型并确定其优先级。此外，您的竞价型分配策略必须是 `price-capacity-optimized`、`capacity-optimized` 或 `lowest-price`。要使用 `capacity-optimized-prioritized` 分配策略，您必须手动添加实例类型并确定其优先级。

使用启动模板创建 Auto Scaling 组

在创建 Auto Scaling 组时，您必须指定配置 Amazon EC2 实例所需的信息、实例的可用区和 VPC 子网、所需容量以及最小和最大容量限制。

要配置通过 Auto Scaling 组启动的 Amazon EC2 实例，您可以指定启动模板或启动配置。以下过程演示如何使用启动模板创建 Auto Scaling 组。

要在创建组后更新 EC2 实例的配置，您可以创建新版本的启动模板。更改 Auto Scaling 组的启动模板后，将使用新的配置选项启动所有新实例，但现有实例不受影响。要更新现有实例，请终止这些实例，以便 Auto Scaling 组替换这些实例，或者根据您的终止策略 (p. 222)启用自动扩展，以逐步使用较新实例替换较旧实例。

> **Note**
>
> 您还可以替换 Auto Scaling 组中的所有实例，以启动使用新启动配置的新实例。有关更多信息，请参阅替换 Auto Scaling 实例 (p. 94)。

先决条件

- 你必须已创建启动模板。有关更多信息，请参阅为 Auto Scaling 组创建启动模板 (p. 20)。
- 您必须拥有 IAM 权限才能使用启动模板创建 Auto Scaling 组。使用启动模板时会检查您的 `ec2:RunInstances` 权限。如果启动模板指定了 IAM 角色，也会检查您的 `iam:PassRole` 权限。有关更多信息，请参阅启动模板支持 (p. 328)。

使用启动模板创建 Auto Scaling 组（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 在屏幕顶部的导航栏中，选择在创建启动模板时使用的同一 Amazon Web Services 区域。
3. 选择 Create an Auto Scaling group (创建 Auto Scaling 组)。
4. 在选择启动模板或配置页面上，执行以下操作：
   a. 在 Auto Scaling 组名称中，输入 Auto Scaling 组的名称。
   b. 对于启动模板，请选择现有启动模板。
   c. 对于 Launch template version（启动模板版本），选择 Auto Scaling 组在扩展时使用启动模板的默认版本、最新版本还是特定版本。
   d. 验证您的启动模板是否支持您计划使用的所有选项，然后选择下一步。
5. 在 Choose instance launch options (选择实例启动选项) 页面的 Network (网络) 下方，对于 VPC，选择相应的 VPC。必须在您于启动模板中指定的安全组所在的 VPC 中创建 Auto Scaling 组。

6. 对于 (子网) Availability Zones and subnets (可用区和子网)，选择指定 VPC 中的一个或多个子网。可以在多个可用区中使用子网以提供高可用性。有关更多信息，请参阅选择 VPC 子网时的注意事项 (p. 302)。
7. 如果您创建了指定实例类型的启动模板，则可以继续下一步，创建使用启动模板中实例类型的 Auto Scaling 组。

   或者，如果在启动模板中未指定实例类型，或者您想使用多种实例类型进行自动扩展，您也可以选择 Override launch template (覆盖启动模板) 选项。有关更多信息，请参阅Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。
8. 选择 Next (下一步) 以继续下一步。

   或者，您可接受其余默认值，然后选择 Skip to review (跳到审核)。
9. （可选）在 Configure advanced options（配置高级选项）页面上，配置以下选项，然后选择 Next（下一步）：

   a. 要向负载均衡器注册您的 Amazon EC2 实例，请选择现有负载均衡器或创建新负载均衡器。有关更多信息，请参阅使用 Elastic Load Balancing 跨 Auto Scaling 组中的实例分配流量。 (p. 272)。要创建新负载均衡器，请按照 从 Amazon EC2 Auto Scaling 控制台配置 Application Load Balancer 或 Network Load Balancer。 (p. 276) 中的过程操作。

   b. 要启用 Elastic Load Balancing (ELB) 运行状况检查，对于运行状况检查，请选择 Health check type（运行状况检查类型）下的 ELB。启用负载均衡时，这些运行状况检查是可选的。

   c. 在 Health check grace period（运行状况检查宽限期）下，输入在实例进入 `InService`状态后，Amazon EC2 Auto Scaling 开始检查其 Elastic Load Balancing 运行状况之前需要的时间。有关更多信息，请参阅设置自动扩缩组的运行状况检查宽限期 (p. 246)。

   d. 在 Additional settings (其他设置)、Monitoring (监控) 下，选择是否启用 CloudWatch 组指标收集。这些指标提供的测量值可以指示潜在的问题，例如终止实例的数量或挂起实例的数量。有关更多信息，请参阅监控 Auto Scaling 组和实例的 CloudWatch 指标 (p. 248)。

   e. 对于 Enable default instance warmup（启用原定设置实例预热），选择此选项并选择应用程序的预热时间。如果您正在创建具有扩缩策略的 Auto Scaling 组，则原定设置的实例预热功能可优化用于动态扩缩的 Amazon CloudWatch 指标。有关更多信息，请参阅为 Auto Scaling 组设置原定设置实例预热 (p. 150)。
10. （可选）在 Configure group size and scaling policies (配置组大小和扩展策略) 页面上，配置以下选项，然后选择 Next (下一步)：

    a. 对于 Desired capacity (所需容量)，请输入要启动的实例的初始数量。将此数字更改为超出最小容量或最大容量限制的值时，必须更新最小容量或最大容量的值。有关更多信息，请参阅为 Auto Scaling 组设置容量限制 (p. 122)。

    b. 要自动缩放 Auto Scaling 组的大小，请选择 Target tracking scaling policy（目标跟踪扩展策略），并按照说明操作。有关更多信息，请参阅Amazon EC2 Auto Scaling 的目标跟踪扩缩策略 (p. 133)。

    c. 在 Instance scale-in protection（实例缩减保护）下，选择是否启用实例缩减保护。有关更多信息，请参阅 实例横向缩减保护 (p. 230)。
11. （可选）要接收通知，请为 Add notification（添加通知）配置通知，然后选择 Next（下一步）。有关更多信息，请参阅在 Auto Scaling 组扩展时获取 Amazon SNS 通知 (p. 258)。
12. （可选）要添加标签，请选择 Add tag（添加标签），为每个标签提供标签键和值，然后选择 Next（下一步）。有关更多信息，请参阅为 Auto Scaling 组和实例添加标签 (p. 87)。
13. 在 Review（查看）页面上，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

使用命令行创建 Auto Scaling 组

您可以使用以下任一命令：

- create-auto-scaling-group (Amazon CLI)
- New-ASAutoScalingGroup (Amazon Tools for Windows PowerShell)

使用启动配置创建 Auto Scaling 组

在创建 Auto Scaling 组时，您必须指定配置 Amazon EC2 实例所需的信息、实例的可用区和 VPC 子网、所需容量以及最小和最大容量限制。

> **Important**
>
> 要配置 Amazon EC2 实例，您可以指定启动模板或启动配置。建议您使用启动模板以确保您可以使用 Amazon EC2 Auto Scaling 和 Amazon EC2 的最新功能。有关更多信息，请参阅使用启动模板创建 Auto Scaling 组 (p. 78)。

以下过程演示如何使用启动配置创建 Auto Scaling 组。您无法在创建启动配置后进行修改，但可以替换 Auto Scaling 组的启动配置。有关更多信息，请参阅更改 Auto Scaling 组的启动配置 (p. 46)。

先决条件

您必须已创建启动配置。有关更多信息，请参阅创建启动配置 (p. 39)。

使用启动配置创建 Auto Scaling 组（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 在屏幕顶部的导航栏中，选择在创建启动配置时使用的同一 Amazon Web Services 区域。
3. 选择 Create an Auto Scaling group (创建 Auto Scaling 组)。
4. 在选择启动模板或配置页面上，对于 Auto Scaling 组名称，输入 Auto Scaling 组的名称。
5. 要选择启动配置，请执行以下操作：
   a. 对于 Launch Template (启动模板)，选择 Switch to launch configuration (切换以启动配置)。
   b. 对于 Launch configuration (启动配置)，请选择现有启动配置。
   c. 验证您的启动配置是否支持您计划使用的所有选项，然后选择 Next (下一步)。
6. 在 (配置设置) Configure instance launch options (配置实例启动选项) 页面的 Network (网络) 下方，对于 VPC，选择相应的 VPC。必须在您于启动配置中指定的安全组所在的 VPC 中创建 Auto Scaling 组。
7. 对于 (子网) Availability Zones and subnets (可用区和子网)，选择指定 VPC 中的一个或多个子网。可以在多个可用区中使用子网以提供高可用性。有关更多信息，请参阅选择 VPC 子网时的注意事项 (p. 302)。
8. 选择 Next（下一步）。

   或者，您可接受其余默认值，然后选择 Skip to review (跳到审核)。
9. （可选）在 Configure advanced options（配置高级选项）页面上，配置以下选项，然后选择 Next（下一步）：
   a. 要向负载均衡器注册您的 Amazon EC2 实例，请选择现有负载均衡器或创建新负载均衡器。有关更多信息，请参阅使用 Elastic Load Balancing 跨 Auto Scaling 组中的实例分配流量。 (p. 272)。要创建新负载均衡器，请按照从 Amazon EC2 Auto Scaling 控制台配置 Application Load Balancer 或 Network Load Balancer。 (p. 276) 中的过程操作。
   b. 要启用 Elastic Load Balancing (ELB) 运行状况检查，对于运行状况检查，请选择 Health check type（运行状况检查类型）下的 ELB。启用负载均衡时，这些运行状况检查是可选的。
   c. 在 Health check grace period（运行状况检查宽限期）下，输入在实例进入 `InService`状态后，Amazon EC2 Auto Scaling 开始检查其 Elastic Load Balancing 运行状况之前需要的时间。有关更多信息，请参阅设置自动扩缩组的运行状况检查宽限期 (p. 246)。
   d. 在 Additional settings (其他设置)、Monitoring (监控) 下，选择是否启用 CloudWatch 组指标收集。这些指标提供的测量值可以指示潜在的问题，例如终止实例的数量或挂起实例的数量。有关更多信息，请参阅监控 Auto Scaling 组和实例的 CloudWatch 指标 (p. 248)。
   e. 对于 Enable default instance warmup（启用原定设置实例预热），选择此选项并选择应用程序的预热时间。如果您正在创建具有扩缩策略的 Auto Scaling 组，则原定设置的实例预热功能可优化用于

动态扩缩的 Amazon CloudWatch 指标。有关更多信息，请参阅为 Auto Scaling 组设置原定设置实例预热 (p. 150)。

10. （可选）在 Configure group size and scaling policies (配置组大小和扩展策略) 页面上，配置以下选项，然后选择 Next (下一步)：

    a. 对于 Desired capacity (所需容量)，请输入要启动的实例的初始数量。将此数字更改为超出最小容量或最大容量限制的值时，必须更新最小容量或最大容量的值。有关更多信息，请参阅为 Auto Scaling 组设置容量限制 (p. 122)。

    b. 要自动缩放 Auto Scaling 组的大小，请选择 Target tracking scaling policy（目标跟踪扩展策略），并按照说明操作。有关更多信息，请参阅Amazon EC2 Auto Scaling 的目标跟踪扩缩策略 (p. 133)。

    c. 在 Instance scale-in protection（实例缩减保护）下，选择是否启用实例缩减保护。有关更多信息，请参阅 实例横向缩减保护 (p. 230)。

11. （可选）要接收通知，请为 Add notification（添加通知）配置通知，然后选择 Next（下一步）。有关更多信息，请参阅在 Auto Scaling 组扩展时获取 Amazon SNS 通知 (p. 258)。

12. （可选）要添加标签，请选择 Add tag（添加标签），为每个标签提供标签键和值，然后选择 Next（下一步）。有关更多信息，请参阅为 Auto Scaling 组和实例添加标签 (p. 87)。

13. 在 Review（查看）页面上，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

使用命令行创建 Auto Scaling 组

您可以使用以下任一命令：

- create-auto-scaling-group (Amazon CLI)
- New-ASAutoScalingGroup (Amazon Tools for Windows PowerShell)

使用现有实例的参数创建 Auto Scaling 组

> **Important**
>
> 此主题仅适用于使用启动配置创建 Auto Scaling 组。我们为尚未从启动配置迁移到启动模板的客户提供有关启动配置的信息。
> 如果这是您第一次创建 Auto Scaling 组，我们建议您使用控制台从现有 EC2 实例创建启动模板。然后使用启动模板创建新的 Auto Scaling 组。有关此步骤，请参阅 使用 Amazon EC2 启动向导创建 Auto Scaling 组 (p. 85)。

以下程序演示了如何通过如下方法创建 Auto Scaling 组：指定要用作启动其他实例基础的现有实例。创建 EC2 实例需要多个参数，例如 Amazon Machine Image (AMI) ID、实例类型、密钥对和安全组。Amazon EC2 Auto Scaling 还使用所有这些信息，以便在需要扩展时代表您启动实例。此信息存储在启动模板或启动配置中。

当您使用现有实例时，Amazon EC2 Auto Scaling 会创建一个 Auto Scaling 组，该组将根据同时创建的启动配置启动实例。Auto Scaling 组的名称与 Auto Scaling 组相同，并且包括来自已识别实例的某些配置详细信息。

以下配置详细信息会从已识别的实例复制到启动配置中：

- AMI ID
- 实例类型
- 密钥对
- 安全组
- IP 地址类型（公有或私有）

- IAM 实例配置文件（如果适用）
- 监控（true 或 false）
- EBS 优化（true 或 false）
- 租期设置 (如果在 VPC (共享或专用) 中启动)
- 内核 ID 和 RAM 磁盘 ID (如果适用)
- 用户数据，如果指定
- Spot（最高）价格

不会从已识别实例中复制以下配置详细信息，

- 存储：不会从已识别的实例中复制块储存设备（EBS 卷和实例存储卷）。相反，作为创建 AMI 的一部分而创建的块储存设备映射决定了使用哪些设备。
- 网络接口数量：网络接口未从已识别的实例中复制。相反，Amazon EC2 Auto Scaling 会使用其默认设置创建一个网络接口，即主网络接口 (eth0)。
- 实例元数据选项：不会从已识别的实例中复制元数据可访问、元数据版本和令牌响应跃点数限制设置。相反，Amazon EC2 Auto Scaling 会使用其默认设置。有关更多信息，请参阅 配置实例元数据选项 (p. 41)。
- 负载均衡器：如果识别的实例适用一个或多个负载均衡器进行注册，则有关负载均衡器的信息不会复制到负载均衡器或新 Auto Scaling 组的目标组属性。
- 标签：如果识别的实例有标签，标签不会复制到新 Auto Scaling 组的 `Tags` 属性。

新的 Auto Scaling 组将实例启动到已识别实例所在的同一可用区、VPC 和子网中。

如果已识别的实例位于置放群组中，则新 Auto Scaling 组将在与已识别实例相同的置放群组中启动实例。由于启动配置设置不允许指定置放群组，因此将置放群组复制到新 Auto Scaling 组的 `PlacementGroup` 属性。

先决条件

EC2 实例必须符合以下标准：

- 实例位于要创建 Auto Scaling 组的子网和可用区中。
- 实例不是其他 Auto Scaling 组的成员。
- 实例处于 `running` 状态。
- 用于启动实例的 AMI 必须仍然存在。

从 EC2 实例创建 Auto Scaling 组 (控制台)

从 EC2 实例创建 Auto Scaling 组 ()

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格上的 Instances（实例）下，选择 Instances（实例），然后选择一个实例。
3. 依次选择操作、实例设置和附加到 Auto Scaling 组。
4. 在附加到 Auto Scaling 组页面上，为 Auto Scaling 组，输入组名称，然后选择附加。

   连接实例后，将其视为 Auto Scaling 组的一部分。将使用新的启动配置和为 Auto Scaling 组指定的相同名称创建新的 Auto Scaling 组。Auto Scaling 组的所需容量和最大容量为 1。
5. （可选）要编辑 Auto Scaling 组的设置，请在导航窗格的 Auto Scaling 下，选择 Auto Scaling Groups（Auto Scaling 组）。选中新 Auto Scaling 组旁边的复选框，选择组列表上方的编辑按钮，根据需要更改设置，然后选择更新。

从 EC2 实例创建 Auto Scaling 组 (Amazon CLI)

在本练习中，我们演示如何使用 Amazon CLI 从 EC2 实例创建 Auto Scaling 组。

此程序不会将实例添加到 Auto Scaling 组中。要连接实例，必须在创建 Auto Scaling 组之后运行 attach-instances 命令。

开始之前，请使用 Amazon EC2 控制台或 describe-instances 命令查找 EC2 实例的 ID。

将当前实例用作模板

- 请使用以下 create-auto-scaling-group 命令从 EC2 实例 `i-0e69cc3f05f825f4f` 创建 Auto Scaling 组，`my-asg-from-instance`。

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg-from-instance \
  --instance-id i-0e69cc3f05f825f4f --min-size 1 --max-size 2 --desired-capacity 2
```

验证 Auto Scaling 组已启动实例

- 请使用以下 describe-auto-scaling-groups 命令验证是否已成功创建 Auto Scaling 组。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg-from-instance
```

以下示例响应显示该组的所需容量为 2，该组有 2 个正在运行的实例，启动配置命名为 `my-asg-from-instance`。

```
{
  "AutoScalingGroups":[
    {
      "AutoScalingGroupName":"my-asg-from-instance",
      "AutoScalingGroupARN":"arn",
      "LaunchConfigurationName":"my-asg-from-instance",
      "MinSize":1,
      "MaxSize":2,
      "DesiredCapacity":2,
      "DefaultCooldown":300,
      "AvailabilityZones":[
        "us-west-2a"
      ],
      "LoadBalancerNames":[ ],
      "TargetGroupARNs":[ ],
      "HealthCheckType":"EC2",
      "HealthCheckGracePeriod":0,
      "Instances":[
        {
          "InstanceId":"i-06905f55584de02da",
          "InstanceType":"t2.micro",
          "AvailabilityZone":"us-west-2a",
          "LifecycleState":"InService",
          "HealthStatus":"Healthy",
          "LaunchConfigurationName":"my-asg-from-instance",
          "ProtectedFromScaleIn":false
        },
        {
          "InstanceId":"i-087b42219468eacde",
          "InstanceType":"t2.micro",
          "AvailabilityZone":"us-west-2a",
          "LifecycleState":"InService",
```

```
          "HealthStatus":"Healthy",
          "LaunchConfigurationName":"my-asg-from-instance",
          "ProtectedFromScaleIn":false
        }
      ],
      "CreatedTime":"2020-10-28T02:39:22.152Z",
      "SuspendedProcesses":[ ],
      "VPCZoneIdentifier":"subnet-6bea5f06",
      "EnabledMetrics":[ ],
      "Tags":[ ],
      "TerminationPolicies":[
        "Default"
      ],
      "NewInstancesProtectedFromScaleIn":false,
      "ServiceLinkedRoleARN":"arn"
    }
  ]
}
```

查看启动配置

- 请使用以下 describe-launch-configurations 命令查看启动配置的详细信息。

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-asg-
from-instance
```

下面是示例输出：

```
{
  "LaunchConfigurations":[
    {
      "LaunchConfigurationName":"my-asg-from-instance",
      "LaunchConfigurationARN":"arn",
      "ImageId":"ami-0528a5175983e7f28",
      "KeyName":"my-key-pair-uswest2",
      "SecurityGroups":[
        "sg-05eaec502fcdadc2e"
      ],
      "ClassicLinkVPCSecurityGroups":[ ],
      "UserData":"",
      "InstanceType":"t2.micro",
      "KernelId":"",
      "RamdiskId":"",
      "BlockDeviceMappings":[ ],
      "InstanceMonitoring":{
        "Enabled":true
      },
      "CreatedTime":"2020-10-28T02:39:22.321Z",
      "EbsOptimized":false,
      "AssociatePublicIpAddress":true
    }
  ]
}
```

终止实例

- 如果您不再需要实例，可终止它。以下 terminate-instances 命令可终止实例i-0e69cc3f05f825f4f。

```
aws ec2 terminate-instances --instance-ids i-0e69cc3f05f825f4f
```

终止 Amazon EC2 实例后，您无法重新启动实例。终止后，卷上的数据都不复存在，并且再也不能附加到任何实例。要了解有关终止实例的更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的终止实例。

使用 Amazon EC2 启动向导创建 Auto Scaling 组

以下程序介绍了如何使用 Amazon EC2 控制台中的启动实例向导创建 Auto Scaling 组。该选项自动使用启动实例向导中的特定配置详细信息来填充启动模板。

Note

该向导不会使用您指定的实例数填充 Auto Scaling 组；它只会使用 Amazon Machine Image (AMI) ID 和实例类型填充启动模板。使用创建 Auto Scaling 组向导以指定要启动的实例数量。AMI 包含配置实例所需的信息。在需要具有相同配置的多个实例时，您可以从单个 AMI 启动多个实例。我们建议使用已在其上安装应用程序的自定义 AMI，以避免在重启属于 Auto Scaling 组的实例时终止实例。要将自定义 AMI 与 Amazon EC2 Auto Scaling 结合使用，您必须首先从自定义实例创建 AMI，然后使用 AMI 为您的 Auto Scaling 组创建启动模板。

先决条件

- 必须在计划于其中创建 Auto Scaling 组的同一个 Amazon Web Services 区域 中创建自定义 AMI。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的创建 AMI
- 您必须拥有 IAM 权限才能使用启动模板创建 Auto Scaling 组并为这些实例创建 EC2 资源。有关更多信息，请参阅 启动模板支持 (p. 328)。

使用自定义 AMI 作为模板

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏上，将显示当前 Amazon Web Services 区域。选择要在其中启动 Auto Scaling 组的区域。
3. 在导航窗格中，选择实例。
4. 选择 Launch instance（启动实例），然后执行以下操作：
   a. 在 Name and tags（名称和标签）下，将 Name（名称）留空。该名称不是用于创建启动模板的数据的一部分。
   b. 在 Application and OS Images (Amazon Machine Image) [应用程序和操作系统镜像（Amazon Machine Image）] 下，选择 Browse more AMIs（浏览其他 AMI）以浏览完整的 AMI 目录。
   c. 选择 My AMIs（我的 AMI），找到您创建的 AMI，然后选择 Select（选择）。
   d. 在 Instance type（实例类型）下，选择一个实例类型。

      Note

      选择创建 AMI 时使用的相同实例类型或更强大的实例类型。

   e. 在屏幕右侧的 Summary（摘要）下，对于 Number of instances（实例的数量），输入一个任意数字。此处输入的数字并不重要。在创建 Auto Scaling 组时，您将会指定要启动的实例数。

      在 Number of instances（实例的数量）字段下，将显示一条消息，其中说明 When launching more than 1 instance, consider EC2 Auto Scaling（启动多个实例时，请考虑 EC2 Auto Scaling）。
   f. 选择 consider EC2 Auto Scaling（考虑 EC2 Auto Scaling）超链接文本。

g. 在 Launch into Auto Scaling Group（启动至 Auto Scaling 组）确认对话框中，选择 Continue（继续）以转至 Create launch template（创建启动模板）页面，其中已经填充了您在启动实例向导中选择的 AMI 和实例类型。

创建启动模板

选择 Continue (继续) 之后，Create launch template (创建启动模板) 页面随即打开。按照此程序创建启动模板。有关更多信息，请参阅 创建启动模板（控制台）(p. 20)。

创建启动模板

1. 在 Launch template name and description（启动模板名称和说明）下，输入新启动模板的名称和描述。
2. （可选）在 Key pair (login) [密钥对（登录）] 下，对于 Key pair name（密钥对名称），选择先前创建以在连接到实例时（例如使用 SSH）使用的密钥对的名称。
3. （可选）在 Network settings（网络设置）下，对于 Security groups（安全组），选择一个或多个之前创建的安全组。
4. （可选）在 Configure storage（配置存储）下，更新存储配置。默认存储配置由 AMI 和实例类型确定。
5. 配置完启动模板后，选择 Create launch template (创建启动模板)。
6. 在确认页面上，选择 Create Auto Scaling group (创建 Auto Scaling 组)。

创建 自动扩缩组

Note

本主题的其余内容介绍创建 Auto Scaling 组的基本步骤。有关可以为 Auto Scaling 组配置的参数的更多说明，请参阅 使用启动模板创建 Auto Scaling 组 (p. 78)。

选择 Create Auto Scaling group (创建 Auto Scaling 组) 之后，Create Auto Scaling group (创建 Auto Scaling 组) 向导随即打开。按照程序创建 Auto Scaling 组。

创建 Auto Scaling 组

1. 在 Choose launch template or configuration (选择启动模板或配置) 页面上，输入 Auto Scaling 组的名称。
2. 已经自动选择您创建的启动模板。

   对于 Launch template version（启动模板版本），选择 Auto Scaling 组在扩展时使用启动模板的默认版本、最新版本还是特定版本。
3. 选择 Next (下一步) 以继续下一步。
4. 在 Choose instance launch options (选择实例启动选项) 页面的 Network (网络) 下方，对于 VPC，选择相应的 VPC。必须在您于启动模板中指定的安全组所在的 VPC 中创建 Auto Scaling 组。

   Tip

   如果您没有在启动模板中指定安全组，则将使用指定 VPC 中的默认安全组启动您的实例。预设情况下，此安全组不允许来自外部网络的入站流量。
5. 对于 (子网) Availability Zones and subnets (可用区和子网)，选择指定 VPC 中的一个或多个子网。
6. 选择 Next (下一步) 两次以转至 Configure group size and scaling policies (配置组大小和扩展策略) 页面。
7. 在 Group size (组大小) 下方，定义 Desired capacity (所需容量)（Auto Scaling 组创建后要立即启动的初始实例数）、Minimum capacity (最小容量)（最小实例数）以及 Maximum capacity (最大容量)（实例的最大数量）。
8. 选择 Skip to review (跳转以查看)。

9. 在 Review（查看）页面上，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

后续步骤

您可以通过查看活动历史记录来检查是否已正确创建了 Auto Scaling 组。在 Activity (活动) 选项卡上的 Activity history (活动历史记录) 下，Status (状态) 列显示您的 Auto Scaling 组是否已成功启动实例。如果实例无法启动，或者它们启动但随后立即终止，请参阅以下主题了解可能的原因和解决方法：

- Amazon EC2 Auto Scaling 问题排查：EC2 实例启动失败 (p. 343)
- Amazon EC2 Auto Scaling 问题排查：AMI 问题 (p. 347)
- Amazon EC2 Auto Scaling 问题排查：运行状况检查 (p. 351)

如果需要，您现在可以在 Auto Scaling 组所在的同一区域中连接负载均衡器。有关更多信息，请参阅 使用 Elastic Load Balancing 跨 Auto Scaling 组中的实例分配流量。 (p. 272)。

为 Auto Scaling 组和实例添加标签

标签是您分配的自定义属性标签，或者是 Amazon 分配给 Amazon 资源的标签。每个标签具有两个部分：

- 标签键（例如，`costcenter`、`environment` 或 `project`）
- 一个称为标签值的可选字段（例如，`111122223333` 或 `production`）

标签可帮助您：

- 跟踪您的 Amazon 成本。您可以在 Amazon Billing and Cost Management 控制面板上激活这些标签。Amazon 使用标签对您的成本进行分类，并向您提供每月成本分配报告。有关更多信息，请参阅 Amazon Billing 用户指南中的使用成本分配标签。
- 根据标签控制对 Auto Scaling 组的访问。您可以使用 IAM policy 中的条件根据该组上的标签控制对自动扩缩组的访问。有关更多信息，请参阅安全性标签 (p. 91)。
- 根据您添加的标签筛选和搜索自动扩缩组。有关更多信息，请参阅使用标签筛选 Auto Scaling 组 (p. 92)。
- 标识和整理您的 Amazon 资源。许多 Amazon Web Services都支持添加标签，因此，您可以将同一标签分配给来自不同服务的资源，以表明这些资源相互之间存在关系。

您可以标记新的或现有 Auto Scaling 组。您还可以将来自某个自动扩缩组的标签传播到该组启动的 EC2 实例。

标签不会传播到 Amazon EBS 卷。要向 Amazon EBS 卷添加标签，请在启动模板中指定标签。有关更多信息，请参阅 为 Auto Scaling 组创建启动模板 (p. 20)。

您可以通过 Amazon Web Services Management Console、Amazon CLI 或 SDK 创建并管理标签。

目录

- 标签命名和使用限制 (p. 88)
- EC2 实例标签生命周期 (p. 88)
- 标记 Auto Scaling 组 (p. 88)
- 删除标签 (p. 90)
- 安全性标签 (p. 91)
- 控制对标签的访问 (p. 91)

- 使用标签筛选 Auto Scaling 组 (p. 92)

标签命名和使用限制

下面是适用于标签的基本限制：

- 每个资源的最大标签数是 50。
- 可以使用单个调用添加或删除的标签的最大数目为 25。
- 最大键长度为 128 个 Unicode 字符。
- 最大值长度为 256 个 Unicode 字符。
- 标签键和值区分大小写。最佳实践是，决定利用标签的策略并在所有资源类型中一致地实施该策略。
- 请勿在标签名称或值中使用 `aws:` 前缀，因为它专为 Amazon 使用预留。您不能编辑或删除具有此前缀的标签名称或值，它们不计入每个资源配额的标签数限制。

EC2 实例标签生命周期

如果您选择将标签传播到 EC2 实例，则将按以下方式管理标签：

- 当 Auto Scaling 组启动实例时，它会在资源创建期间向实例添加标记，而不是在创建资源之后。
- Auto Scaling 组会自动向实例添加带有 `aws:autoscaling:groupName` 键和 Auto Scaling 组名称值的标签。
- 如果您在启动模板中指定了实例标签，并且选择将组的标签传播到其实例，则所有标签都会合并。如果为启动模板中的标签和 Auto Scaling 组中的标签指定了相同的标签键，则优先使用该组中的标签值。
- 您在附加现有实例时，Auto Scaling 组就会向这些实例添加标签，覆盖具有相同标签关键字的现有标签。它还添加键为 `aws:autoscaling:groupName`、值为 Auto Scaling 组名称的标签。
- 将实例从 Auto Scaling 组中分离时，它仅删除 `aws:autoscaling:groupName` 标签。

标记 Auto Scaling 组

当您向 Auto Scaling 组添加标签时，可以指定是否应将其添加到 Auto Scaling 组中启动的实例。如果修改标签，在更改后，标签的更新版本将添加到在 Auto Scaling 组中启动的实例。如果创建或修改 Auto Scaling 组的标签，不会对已经在 Auto Scaling 组中运行的实例进行这些更改。

目录

- 添加或修改标签（控制台）(p. 88)
- 添加或修改标签 (Amazon CLI) (p. 89)

添加或修改标签（控制台）

创建时标记 Auto Scaling 组

当您使用 Amazon EC2 控制台创建 Auto Scaling 组时，您可以在“创建 Auto Scaling 组”向导的添加标签页面上指定标签键和值。要将标签传播到在 Auto Scaling 组中启动的实例，请确保该标签的 Tag new instances（标记新实例）选项保持选中状态。否则，您可以取消选择它。

添加或修改现有 Auto Scaling 组的标签

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。

2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details（详细信息）选项卡上，选择 Tags（标签）、Edit（编辑）。
4. 要修改现有标签，请编辑 Key（键）和 Value（值）。
5. 要添加新标签，请选择 Add tag（添加标签），然后编辑 Key（键）和 Value（值）。您可以使标记新实例保持选中状态，以便自动将标签添加到在 Auto Scaling 组启动的实例，否则取消选中它。
6. 添加完标签后，选择 Update（更新）。

添加或修改标签 (Amazon CLI)

以下示例说明如何使用 Amazon CLI 在您创建 Auto Scaling 组时添加标签，以及添加或修改现有 Auto Scaling 组的标签。

创建时标记 Auto Scaling 组

使用 create-auto-scaling-group 命令创建新的 Auto Scaling 组并将标签（例如 **environment=production**）添加到 Auto Scaling 组。该标签还会添加到在 Auto Scaling 组中启动的任何实例。

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-configuration-name my-launch-config --min-size 1 --max-size 3 \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --tags Key=environment,Value=production,PropagateAtLaunch=true
```

创建或修改现有 Auto Scaling 组的标签

使用 create-or-update-tags 命令创建或修改标签。例如，以下命令将添加 **Name=my-asg** 和 **costcenter=cc123** 标签。在进行该更改后，该标签还会添加到在 Auto Scaling 组中启动的任何实例。如果具有任一键的标签已经存在，则会替换现有标签。Amazon EC2 控制台将每个实例的显示名称与为 Name 键指定的名称（区分大小写）关联。

```
aws autoscaling create-or-update-tags \
  --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Name,Value=my-
asg,PropagateAtLaunch=true \
  ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=costcenter,Value=cc123,PropagateAtLaunch=true
```

描述 Auto Scaling 组的标签 (Amazon CLI)

如果您要查看应用于特定的 Auto Scaling 组的标签，可以使用以下任一命令：

- describe-tags — 提供 Auto Scaling 组的名称，以查看指定组的标签列表。

  ```
  aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
  ```

  以下为响应示例。

  ```
  {
      "Tags": [
          {
              "ResourceType": "auto-scaling-group",
              "ResourceId": "my-asg",
              "PropagateAtLaunch": true,
              "Value": "production",
  ```

```
            "Key": "environment"
        }
    ]
}
```

- describe-auto-scaling-groups — 提供 Auto Scaling 组名称以查看指定组的属性，包括任何标签。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

以下为响应示例。

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupARN": "arn",
            "HealthCheckGracePeriod": 0,
            "SuspendedProcesses": [],
            "DesiredCapacity": 1,
            "Tags": [
                {
                    "ResourceType": "auto-scaling-group",
                    "ResourceId": "my-asg",
                    "PropagateAtLaunch": true,
                    "Value": "production",
                    "Key": "environment"
                }
            ],
            "EnabledMetrics": [],
            "LoadBalancerNames": [],
            "AutoScalingGroupName": "my-asg",
            ...
        }
    ]
}
```

删除标签

您可以随时删除与 Auto Scaling 组关联的标签。

目录

- 删除标签（控制台）(p. 90)
- 删除标签 (Amazon CLI) (p. 91)

删除标签（控制台）

删除标签

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中现有组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details（详细信息）选项卡上，选择 Tags（标签）、Edit（编辑）。
4. 选择标签旁边的 Remove（删除）。

5. 选择 Update（更新）。

删除标签 (Amazon CLI)

使用 delete-tags 命令删除标签。例如，以下命令删除键为 **environment** 的标签。

```
aws autoscaling delete-tags --tags "ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=environment"
```

您必须指定标签键，但无需指定值。如果您指定了一个值，并且该值不正确，则不会删除标签。

安全性标签

使用标签来验证请求者（例如 IAM 用户或角色）是否有权创建、修改或删除特定自动扩缩组。使用下面的一个或多个条件键，在 IAM policy 的条件元素中提供标签信息：

- 使用 `autoscaling:ResourceTag/tag-key: tag-value` 可允许（或拒绝）带特定标签的 Auto Scaling 组上的用户操作。
- 使用 `aws:RequestTag/tag-key: tag-value` 要求在请求中存在（或不存在）特定标签。
- 使用 `aws:TagKeys [tag-key, ...]` 要求在请求中存在（或不存在）特定标签键。

例如，您可能拒绝对包含具有键 **environment** 和值 **production** 的标签的 Auto Scaling 组的访问，如以下示例所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:DeleteAutoScalingGroup"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {"autoscaling:ResourceTag/environment": "production"}
            }
        }
    ]
}
```

有关更多示例，请参阅 Amazon EC2 Auto Scaling 基于身份的策略示例 (p. 319)：

控制对标签的访问

使用标签来验证请求者（例如 IAM 用户或角色）是否有权添加、修改或删除自动扩缩组的标签。

例如，您可以创建一个 IAM policy，以仅允许从自动扩缩组删除具有 **temporary** 键的标签。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
                "autoscaling:DeleteTags"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": { "aws:TagKeys": ["temporary"] }
            }
        }
    ]
}
```

有关更多示例，请参阅 Amazon EC2 Auto Scaling 基于身份的策略示例 (p. 319)：

Note

在实例启动后，即使您制定限制您的用户对 Auto Scaling 组执行标记（或取消标记）操作的策略，这也不会防止他们手动更改实例上的标签。有关控制对 EC2 实例上标签访问的示例，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的示例：标记资源。

使用标签筛选 Auto Scaling 组

以下示例向您演示如何将筛选条件与 describe-auto-scaling-groups 命令搭配使用来描述具有特定标签的 Auto Scaling 组。按标签筛选仅限于 Amazon CLI 或 SDK，并且不可从控制台使用。

筛选注意事项

- 您可以在单一请求中指定多个筛选条件和多个筛选条件值。
- 您不可以将通配符与筛选值一同使用。
- 筛选值区分大小写。

示例：使用特定标签键和值对描述 Auto Scaling 组

以下命令展示了如何筛选结果以仅显示具有 **environment=production** 的标签键和值对的 Auto Scaling 组。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-key,Values=environment Name=tag-value,Values=production
```

以下为响应示例。

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupARN": "arn",
            "HealthCheckGracePeriod": 0,
            "SuspendedProcesses": [],
            "DesiredCapacity": 1,
            "Tags": [
                {
                    "ResourceType": "auto-scaling-group",
                    "ResourceId": "my-asg",
                    "PropagateAtLaunch": true,
                    "Value": "production",
                    "Key": "environment"
                }
            ],
            "EnabledMetrics": [],
            "LoadBalancerNames": [],
```

```
            "AutoScalingGroupName": "my-asg",
            ...
        }
    ]
}
```

或者，您也可以使用 `tag:<key>` 筛选条件指定标签。例如，以下命令展示了如何筛选结果以仅显示具有 **environment=production** 的标签键和值对的 Auto Scaling 组。此筛选条件的格式如下所示：`Name=tag:<key>,Values=<value>`，其中采用代表标签键值对的 **<key>** 和 **<value>**。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag:environment,Values=production
```

你也可以通过使用 `--query` 选项筛选 Amazon CLI 输出。以下示例演示了如何限制上一条命令的 Amazon CLI 输出仅限为仅给出组名称、最小容量、最大容量和所需容量属性。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag:environment,Values=production \
  --query "AutoScalingGroups[].{AutoScalingGroupName: AutoScalingGroupName, MinSize:
 MinSize, MaxSize: MaxSize, DesiredCapacity: DesiredCapacity}"
```

以下为响应示例。

```
[
    {
        "AutoScalingGroupName": "my-asg",
        "MinSize": 0,
        "MaxSize": 10,
        "DesiredCapacity": 1
    }
  ...
]
```

有关筛选的详细信息，请参阅 Amazon Command Line Interface 用户指南中的筛选 Amazon CLI 输出。

示例：描述带有与指定标签键匹配的标签的 Auto Scaling 组

以下命令演示了如何筛选结果以仅显示带有 **environment** 标签的 Auto Scaling 组，并且不考虑标签值。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-key,Values=environment
```

示例：描述带有与指定标签键集匹配的标签的 Auto Scaling 组

以下命令显示如何筛选结果以仅显示带有 **environment** 和 **project** 的标签的 Auto Scaling 组，并且不考虑标签值。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-key,Values=environment Name=tag-key,Values=project
```

示例：描述具有与至少一个指定标签键匹配的标签的 Auto Scaling 组

以下命令显示如何筛选结果以仅显示带有 **environment** 或 **project** 的标签的 Auto Scaling 组，并且不考虑标签值。

```
aws autoscaling describe-auto-scaling-groups \
```

```
  --filters Name=tag-key,Values=environment,project
```

示例：描述带有指定标签值的 Auto Scaling 组

以下命令展示了如何筛选结果以仅显示标签值为 **production** 的 Auto Scaling 组，并且不考虑标签键。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-value,Values=production
```

示例：描述带有指定标签值集的 Auto Scaling 组

以下命令显示如何筛选结果以仅显示具有标签值 **production** 和 **development** 的 Auto Scaling 组，并且不考虑标签键。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-value,Values=production Name=tag-value,Values=development
```

示例：描述带有与至少一个指定标签值匹配的标签的 Auto Scaling 组

以下命令展示了如何筛选结果以仅显示标签值为 **production** 或**development** 的 Auto Scaling 组，并且不考虑标签键。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-value,Values=production,development
```

示例：描述带有与多个标签键和值匹配的标签的 Auto Scaling 组

您还可以组合筛选条件来创建自定义 AND 和 OR 逻辑，以执行更复杂的筛选。

以下命令显示如何筛选结果以仅显示具有特定标签集的 Auto Scaling 组。一个标签键是 **environment** AND，并且标签值是 (**production** OR **development**) AND；而另一个标签键是 **costcenter** AND，并且标签值是 **cc123**。

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag:environment,Values=production,development
 Name=tag:costcenter,Values=cc123
```

替换 Auto Scaling 实例

Amazon EC2 Auto Scaling 提供的功能允许您在更新或修改 Auto Scaling 组后替换实例。Amazon EC2 Auto Scaling 还可以让您选择将更新包含在替换实例的同一操作中，从而帮助您简化更新。

本部分包含有助于您实现以下功能的信息：

- 启动实例刷新以替换 Auto Scaling 组中的实例。
- 声明描述所需配置的特定更新，并将 Auto Scaling 组更新为所需配置。
- 跳过替换已更新的实例。
- 现在，您可以将检查点添加到实例刷新中，以分阶段替换实例并在特定时间点对您的实例执行验证。
- 达到检查点时通过电子邮件接收通知。
- 使用回滚将自动扩缩组恢复到其以前使用的配置。
- 限制实例的生命周期，以便在整个自动扩缩组中提供一致的软件版本和实例配置。

目录

- 基于实例刷新替换 Auto Scaling 实例 (p. 95)
- 基于最大实例生命周期替换 Auto Scaling 实例 (p. 115)

基于实例刷新替换 Auto Scaling 实例

您可以使用实例刷新更新 Auto Scaling 组中的实例，而不是一次手动替换几个实例。如果配置更改要求您替换实例，并且您的 Auto Scaling 组中有大量实例，则此功能非常有用。

如果您有新 Amazon Machine Image (AMI) 或新的用户数据脚本，则实例刷新可能非常有用。要使用实例刷新，请首先创建一个新的启动模板，该模板指定新的 AMI 或用户数据脚本。然后，启动实例刷新以立即开始更新组中的实例。

目录

- 工作原理 (p. 95)
- 核心概念和术语 (p. 95)
- 实例类型兼容性 (p. 96)
- 限制 (p. 97)
- 启动实例刷新 (p. 97)
- 了解实例刷新的默认值 (p. 101)
- 检查实例刷新的状态 (p. 103)
- 取消实例刷新 (p. 104)
- 通过回滚撤消更改 (p. 105)
- 使用实例刷新和跳过匹配 (p. 107)
- 将检查点添加到实例刷新 (p. 112)

工作原理

下面的示例步骤显示了您如何使用实例刷新来更新带有新的启动模板的自动扩缩组：

- 为自动扩缩组创建一个新的启动模板。有关更多信息，请参阅为 Auto Scaling 组创建启动模板 (p. 20)。
- 您可以配置最低运行正常百分比、实例预热和检查点，指定包含启动模板的所需配置，然后开启实例刷新。所需的配置可以选择指定是否应用混合实例策略 (p. 50)。
- Amazon EC2 Auto Scaling 开始执行实例的滚动替换。它使一组实例停止服务，终止它们，然后启动一组具有新所需配置的实例。然后，它会等到实例通过您的运行状况检查并完成预热，然后再进入替换其他实例。
- 替换一定百分比的组后，会达到检查点。无论何时有检查点，Amazon EC2 Auto Scaling 都会临时停止替换实例，并发送通知。然后，它会等待您指定的时间，然后再继续。收到通知后，您可以验证您的新实例是否正常工作。
- 实例刷新成功后，Auto Scaling 组设置将使用您在操作开始时指定的配置自动更新。

核心概念和术语

在您开始之前，请熟悉以下实例刷新核心概念和术语：

最低运行正常百分比

作为启动实例刷新的一部分，您可以指定始终保持的最低运行正常百分比。这是自动扩缩组中在实例刷新期间必须通过运行状况检查 (p. 241)的容量数，以便刷新继续执行。例如，如果最小运行状况百分

比为 90%，那么 10% 即是将要终止和替换的容量百分比。如果新的实例未通过运行状况检查，Amazon EC2 Auto Scaling 将终止并替换这些实例。如果它无法启动任何运行正常的实例，则实例刷新最终将失败，使组中其他的 90% 保持不变。如果新的实例保持正常运行并已完成预热期，Amazon EC2 Auto Scaling 可以继续替换其他实例。

实例刷新可以一次替换一个实例，一次替换多个实例，也可以一次替换全部实例。要一次替换一个实例，将最低运行正常百分比设置为 100%。要一次替换全部实例，将最低运行正常百分比设置为 0%。

Amazon EC2 Auto Scaling 会根据您的自动扩缩组所使用的运行状况检查的状态来确定实例是否运行正常。有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。为确保 Amazon EC2 Auto Scaling 能够尽快检测到实例的运行状况问题，请勿将组的运行状况检查宽限期设置得过高。有关更多信息，请参阅设置自动扩缩组的运行状况检查宽限期 (p. 246)。

实例预热

Important

使用默认实例预热来定义自动扩缩组的预热周期。只有在必须覆盖默认实例预热时，才为实例刷新指定实例预热。实例预热与默认实例预热的工作方式相同，因此实例预热适用相同的扩展注意事项。有关更多信息，请参阅为 Auto Scaling 组设置原定设置实例预热 (p. 150)。

实例预热是从新实例的状态变为 `InService` 之时起到其可接收流量之时止需要经过的时间段。在实例刷新期间，如果实例通过其运行状况检查，确定新启动的实例正常运行后，Amazon EC2 Auto Scaling 不会立即继续进行下一个实例替换。它会等待预热周期，然后继续替换下一个实例。如果应用程序在开始提供流量之前需要时间完成自身的初始化，此功能可能很有用。

如果您使用生命周期挂钩准备新的实例以供使用，则可以降低预热期的值。有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

所需配置

所需配置是您希望 Amazon EC2 Auto Scaling 在自动扩缩组中部署的新配置。例如，您可以为实例指定新的启动模板和新的实例类型。在实例刷新期间，Amazon EC2 Auto Scaling 会将 Auto Scaling 组更新为所需配置。如果在实例刷新期间发生向外扩展事件，Amazon EC2 Auto Scaling 将启动具有所需配置而不是该组的当前设置的新实例。实例刷新成功后，Amazon EC2 Auto Scaling 会更新自动扩缩组的设置，以反映您在实例刷新过程中指定的新所需配置。

跳过匹配

跳过匹配将通知 Amazon EC2 Auto Scaling 忽略已有最新更新的实例。这样，您就无需替换超过所需数量的实例。当您希望确保自动扩缩组使用特定版本的启动模板，并且只替换那些使用其他版本的实例时，这是很有用的。

检查点

检查点是实例刷新在指定时间内暂停的时间点。实例刷新可以包含多个检查点。Amazon EC2 Auto Scaling 会针对每个检查点发出事件。因此，您可以添加 EventBridge 规则，将事件发送到目标（如 Amazon SNS），以便在达到检查点时收到通知。达到某个检查点后，您将有机会验证您的部署。如果发现任何问题，您可以取消实例刷新，或回滚实例。分阶段部署更新的能力是检查点的一个关键优势。如果不使用检查点，则会持续执行滚动替换。

实例类型兼容性

在更改实例类型之前，最好验证它是否适用于您的启动模板。这确认了与您指定的 AMI 的兼容性。例如，如果您从半虚拟化（PV）AMI 启动了原始实例，但希望更改为仅受硬件虚拟机（HVM）AMI 支持的最新一代实例类型。在这种情况下，您必须在启动模板中使用 HVM AMI。

要在不启动实例的情况下确认实例类型的兼容性，请使用带 `--dry-run` 选项的 run-instances 命令，如以下示例所示。

```
aws ec2 run-instances --launch-template LaunchTemplateName=my-template,Version='1' --dry-
run
```

有关如何确定兼容性的信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的更改实例类型的兼容性。

限制

- Instances terminated before launch（实例在启动前终止）：当 Auto Scaling 组中只有一个实例时，开启实例刷新可能会导致中断。这是因为 Amazon EC2 Auto Scaling 会终止实例，然后启动新实例。
- 总持续时间：实例刷新可继续主动替换实例的最长时间为 14 天。
- 加权组特有的行为差异：如果配置混合实例组的实例权重大于或等于该组的所需容量，则 Amazon EC2 Auto Scaling 可能会一次性替换所有 `InService` 实例。为避免这种情况，请遵循 为 Amazon EC2 Auto Scaling 配置实例权重 (p. 67) 主题中的建议。指定所需容量，该容量大于在自动扩缩组中使用权重时的最大权重。
- 一小时超时：当实例刷新因等待替换备用实例或横向缩减保护实例，或者新实例没有通过运行状况检查而无法继续进行替换时，Amazon EC2 Auto Scaling 将持续重试一个小时。它还将提供状态消息以帮助您解决问题。如果问题在一小时后仍然存在，则操作失败。这样做的目的是在出现临时问题时给它时间恢复。

启动实例刷新

> **Important**
>
> 您可以回滚正在进行的实例刷新，以撤消任何更改。要使此功能起作用，自动扩缩组在启动实例刷新之前必须满足使用回滚的先决条件。有关更多信息，请参阅通过回滚撤消更改 (p. 105)。

以下过程可帮助您使用 Amazon Web Services Management Console 或 Amazon CLI 启动实例刷新。

启动实例刷新（控制台）

如果这是您第一次开启实例刷新，则使用控制台将有助于您了解可用的功能和选项。

在控制台中开启实例刷新（基本步骤）

如果您以前没有为 Auto Scaling 组定义混合实例策略 (p. 50)，请使用下面的步骤。如果您以前定义了混合实例策略，请参阅 在控制台中开启实例刷新（混合实例组）(p. 99) 以开启实例刷新。

启动实例刷新

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling group（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Instance refresh（实例刷新）选项卡上的 Active instance refresh（活跃实例刷新）中，选择 Start instance refresh（开启实例刷新）。
4. 对于 Minimum healthy percentage（最低运行正常百分比），输入在实例刷新期间必须保持健康状态的 Auto Scaling 组的百分比。将最低运行正常百分比设置为 100% 会将替换率限制为一次只能替换一个实例。相比之下，将其设置为 0% 会同时替换所有实例。
5. 对于实例预热，输入从新实例的状态变为 `InService` 之后到其可以接收流量时为止需要经过的秒数。Amazon EC2 Auto Scaling 在继续替换下一个实例之前会等待这段时间。

   预热时，新启动的实例也不会计入自动扩缩组的聚合实例指标（例如 `CPUUtilization`、`NetworkIn` 和 `NetworkOut`）。如果已在 Auto Scaling 组中添加了扩展策略，扩展活动将并行运行。如果为实例刷新设置了较长的预热周期，则新启动的实例需要更多的时间才能显示在指标中。因此，设置充分的预热时间可防止 Amazon EC2 Auto Scaling 根据过时的指标数据进行扩缩。

如果您已经为自动扩缩组正确定义了默认实例预热，则无需更改实例预热。但是，如果您想覆盖默认，则可以。有关设置默认实例预热的更多信息，请参阅 为 Auto Scaling 组设置原定设置实例预热 (p. 150)。

6. （可选）对于 Checkpoints（检查点），选择 Enable checkpoints（启用检查点），以使用递增量或分阶段的实例刷新方法来替换实例。这将为在替换集之间进行验证提供额外时间。如果您选择不启用检查点，将以一次近乎连续的操作替换这些实例。

   如果您启用检查点，请参阅 启用检查点（控制台）(p. 113) 以了解其他步骤。

7. 启用或关闭 Skip matching（跳过匹配）：

   - 若要跳过替换已与您的启动模板匹配的实例，请将启用跳过匹配复选框保持为选中状态。
   - 如果通过清除此复选框关闭跳过匹配，则可替换所有实例。

   当您启用跳过匹配，而不是使用当前启动模板时，则可以设置新启动模板或启动模板的新版本。您可以在启动实例刷新页面的所需配置部分中执行此操作。

   Note

   要使用跳过匹配功能更新当前使用启动配置的 Auto Scaling 组，您必须在 Desired configuration（所需配置）中选择启动模板。不支持跳过匹配与启动配置一起使用。

8. 对于备用实例，选择忽略、终止或等待。这决定了当实例处于 `Standby` 状态时会发生什么。有关更多信息，请参阅临时从 Auto Scaling 组中移除实例 (p. 232)。

   如果您选择等待，则必须采取其他步骤才能使这些实例恢复服务。如果您不这样做，则实例刷新会替换所有 `InService` 实例并等待一小时。然后，如果还有任何 `Standby` 实例，则实例刷新将失败。为防止出现这种情况，请改为选择忽略或终止这些实例。

9. 对于横向缩减保护实例，选择忽略、替换或等待。这决定了如果找到横向缩减保护实例会发生什么。有关更多信息，请参阅实例横向缩减保护 (p. 230)。

   如果您选择等待，则必须采取其他措施来删除这些实例的横向缩减保护。如果您不这样做，则实例刷新会替换所有未受保护的实例并等待一小时。然后，如果还有任何横向缩减保护实例，则实例刷新将失败。为防止出现这种情况，请改为选择忽略或替换这些实例。

10. （可选）展开所需配置部分，以指定您要对自动扩缩组进行的任何更新。

    对于此步骤，您可以选择使用 JSON 或 YAML 语法来编辑参数值，而不是在控制台界面中进行选择。为此，请选择 Use code editor（使用代码编辑器），而不是 Use console interface（使用控制台界面）。以下步骤介绍如何使用控制台界面进行选择。

    a. 对于 Update launch template（更新启动模板）：

       - 如果您尚未为自动扩缩组创建新启动模板或新启动模板版本，则不要选中此复选框。
       - 如果您已创建新启动模板或新启动模板版本，请选中此复选框。当您选择此选项时，Amazon EC2 Auto Scaling 将为您显示当前启动模板和当前启动模板版本。它还列出了任何其他可用版本。选择启动模板，然后选择版本。

         在您选择版本后，可以看到版本信息。这是作为实例刷新的组成部分替换实例时将要使用的启动模板的版本。如果实例刷新成功，则每当新实例启动时（例如在组扩展时），也将使用此版本的启动模板。

    b. 对于 Choose a set of instance types and purchase options to override the instance type in the launch template（选择一组实例类型和购买选项以覆盖启动模板中的实例类型）：

       - 如果您想使用您在启动模板中指定的实例类型和购买选项，请不要选中此复选框。
       - 如果您要覆盖启动模板中的实例类型或运行竞价型实例，请选中此复选框。您可以手动添加每种实例类型，也可以选择主实例类型和为您检索任何其他匹配实例类型的建议选项。如果您计划启动竞价型实例，我们建议添加几种不同的实例类型。这样，如果您选择的可用区中实例容

量不足，则 Amazon EC2 Auto Scaling 可以启动另一种实例类型。有关更多信息，请参阅Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。

Warning

不要将竞价型实例用于无法处理竞价型实例中断的应用程序。如果 Amazon EC2 Spot 服务需要回收容量，可能会发生中断。

如果您选中此复选框，请确保启动模板尚未请求竞价型实例。您无法使用请求竞价型实例的启动模板来创建使用多种实例类型并启动竞价型和按需型实例的自动扩缩组。

Note

要在当前使用启动配置的 Auto Scaling 组上配置这些选项，您必须在 Update launch template（更新启动模板）中选择启动模板。不支持覆盖您的启动配置中的实例类型。

11. （可选）对于回滚设置，选择启用自动回滚以在实例刷新失败时自动回滚实例刷新。

    当自动扩缩组不满足使用回滚的先决条件时，无法启用此设置。

    有关更多信息，请参阅通过回滚撤消更改 (p. 105)。

12. 审核您的所有选择，以确认所有选择的设置都正确。

    此时，最好验证当前更改与建议更改之间的差异不会以意外或不需要的方式影响您的应用程序。要确认您的实例类型与您的启动模板兼容，请参阅 实例类型兼容性 (p. 96)。

13. 如果您对实例刷新选择感到满意，请选择 Start（开启）。

在控制台中开启实例刷新（混合实例组）

如果您已使用混合实例策略 (p. 50)创建了 Auto Scaling 组，则请使用以下步骤。如果您尚未为您的组定义混合实例策略，请参阅 在控制台中开启实例刷新（基本步骤） (p. 97) 以开启实例刷新。

启动实例刷新

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling group（Auto Scaling 组）页面底部打开一个拆分窗格。

3. 在 Instance refresh（实例刷新）选项卡上的 Active instance refresh（活跃实例刷新）中，选择 Start instance refresh（开启实例刷新）。
4. 对于 Minimum healthy percentage（最低运行正常百分比），输入在实例刷新期间必须保持健康状态的 Auto Scaling 组的百分比。将最低运行正常百分比设置为 100% 会将替换率限制为一次只能替换一个实例。相比之下，将其设置为 0% 会同时替换所有实例。
5. 对于实例预热，输入从新实例的状态变为 `InService` 之后到其可以接收流量时为止需要经过的秒数。Amazon EC2 Auto Scaling 在继续替换下一个实例之前会等待这段时间。

   预热时，新启动的实例也不会计入自动扩缩组的聚合实例指标（例如 `CPUUtilization`、`NetworkIn`、`NetworkOut` 等等）。如果已在 Auto Scaling 组中添加了扩展策略，扩展活动将并行运行。如果为实例刷新设置了较长的预热周期，则新启动的实例需要更多的时间才能显示在指标中。因此，设置充分的预热时间可防止 Amazon EC2 Auto Scaling 根据过时的指标数据进行扩缩。

   如果您已经正确定义了自动扩缩组的默认实例预热，则无需更改实例预热（除非您想覆盖默认）。有关设置默认实例预热的更多信息，请参阅 为 Auto Scaling 组设置原定设置实例预热 (p. 150)。

6. （可选）对于 Checkpoints（检查点），选择 Enable checkpoints（启用检查点），以使用递增量或分阶段的实例刷新方法来替换实例。这将为在替换集之间进行验证提供额外时间。如果您选择不启用检查点，将以一次近乎连续的操作替换这些实例。

   如果您启用检查点，请参阅 启用检查点（控制台）(p. 113) 以了解其他步骤。
7. 启用或关闭 Skip matching（跳过匹配）：

   - 若要跳过替换已与您的启动模板匹配的实例和任何实例类型覆盖，请将启用跳过匹配复选框保持为选中状态。
   - 如果您选择通过清除此复选框关闭跳过匹配，则可替换所有实例。

   当您启用跳过匹配，而不是使用当前启动模板时，则可以设置新启动模板或启动模板的新版本。您可以在启动实例刷新页面的所需配置部分中执行此操作。您还可以在 Desired configuration（所需配置）中更新实例类型覆盖。
8. 对于备用实例，选择忽略、终止或等待。这决定了当实例处于 `Standby` 状态时会发生什么。有关更多信息，请参阅临时从 Auto Scaling 组中移除实例 (p. 232)。

   如果您选择等待，则必须采取其他步骤才能使这些实例恢复服务。如果您不这样做，则实例刷新会替换所有 `InService` 实例并等待一小时。然后，如果还有任何 `Standby` 实例，则实例刷新将失败。为防止出现这种情况，请改为选择忽略或终止这些实例。
9. 对于横向缩减保护实例，选择忽略、替换或等待。这决定了如果找到横向缩减保护实例会发生什么。有关更多信息，请参阅实例横向缩减保护 (p. 230)。

   如果您选择等待，则必须采取其他措施来删除这些实例的横向缩减保护。如果您不这样做，则实例刷新会替换所有未受保护的实例并等待一小时。然后，如果还有任何横向缩减保护实例，则实例刷新将失败。为防止出现这种情况，请改为选择忽略或替换这些实例。
10. 在 Desired configuration（所需配置）部分中，执行以下操作。

    对于此步骤，您可以选择使用 JSON 或 YAML 语法来编辑参数值，而不是在控制台界面中进行选择。为此，请选择 Use code editor（使用代码编辑器），而不是 Use console interface（使用控制台界面）。以下步骤介绍如何使用控制台界面进行选择。

    a. 对于 Update launch template（更新启动模板）：

       - 如果您尚未为自动扩缩组创建新启动模板或新启动模板版本，则不要选中此复选框。
       - 如果您已创建新启动模板或新启动模板版本，请选中此复选框。当您选择此选项时，Amazon EC2 Auto Scaling 将为您显示当前启动模板和当前启动模板版本。它还列出了任何其他可用版本。选择启动模板，然后选择版本。

         在您选择版本后，可以看到版本信息。这是作为实例刷新的组成部分替换实例时将要使用的启动模板的版本。如果实例刷新成功，则每当新实例启动时（例如在组扩展时），也将使用此版本的启动模板。

    b. 对于 Use these settings to override the instance type and purchase option defined in the launch template（使用这些设置覆盖在启动模板中定义的实例类型和购买选项）：

       默认情况下，此复选框处于选中状态。Amazon EC2 Auto Scaling 将使用当前在适用于 Auto Scaling 组的混合实例策略中设置的值填充每个参数。仅更新您要更改的参数的值。有关这些设置的指导，请参阅 Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。

       Warning

       我们建议您不要清除此复选框。仅当您希望停止使用混合实例策略时才清除它。在实例刷新成功后，Amazon EC2 Auto Scaling 会将您的组更新为与 Desired configuration（所需配置）匹配。如果它不再包含混合实例策略，Amazon EC2 Auto Scaling 将逐渐终止当前运行的所有竞价型实例，并将它们替换为按需型实例。或者，如果您的启动模板请求竞价型实例，则 Amazon EC2 Auto Scaling 将逐渐终止当前正在运行的所有按需型实例，并将它们替换为竞价型实例。

11. （可选）对于回滚设置，选择启用自动回滚以在实例刷新因任何原因而失败时自动回滚。

    当自动扩缩组不满足使用回滚的先决条件时，无法启用此设置。

    有关更多信息，请参阅通过回滚撤消更改 (p. 105)。
12. 审核您的所有选择，以确认所有选择的设置都正确。

    此时，最好验证当前更改与建议更改之间的差异不会以意外或不需要的方式影响您的应用程序。要确认您的实例类型与您的启动模板兼容，请参阅 实例类型兼容性 (p. 96)。

    如果您对实例刷新选择感到满意，请选择 Start（开启）。

启动实例刷新（Amazon CLI）

启动实例刷新

使用以下 start-instance-refresh 命令从 Amazon CLI 启动实例刷新。您可以在 JSON 配置文件中指定要更改的任何首选项。引用配置文件时，请提供该文件的路径和名称，如以下示例所示。

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

config.json 的内容：

```
{
    "AutoScalingGroupName": "my-asg",
    "Preferences": {
      "InstanceWarmup": 60,
      "MinHealthyPercentage": 50,
      "AutoRollback": true,
      "ScaleInProtectedInstances": Ignore,
      "StandbyInstances": Terminate
    }
}
```

如果不提供首选项，则会使用默认值。有关更多信息，请参阅了解实例刷新的默认值 (p. 101)。

输出示例：

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

了解实例刷新的默认值

您可以使用 Amazon Web Services Management Console 或命令行中的配置文件自定义影响实例刷新的各种首选项。一些首选项默认会有所不同，具体取决于您使用的是控制台还是命令行。

当您开始刷新实例时，您可以更改以下任何首选项：

自动回滚（**AutoRollback**）

选择在实例刷新失败时是否将自动扩缩组回滚到其先前的配置。有关更多信息，请参阅通过回滚撤消更改 (p. 105)。

检查点（**CheckpointPercentages**）

选择是否分阶段替换实例。这样，您可以随时对实例执行验证。有关更多信息，请参阅将检查点添加到实例刷新 (p. 112)。

检查点延迟（**CheckpointDelay**）

在到达检查点之后与继续操作之前需要等待的时间量（以秒为单位）。有关更多信息，请参阅将检查点添加到实例刷新 (p. 112)。

实例预热（**InstanceWarmup**）

以秒为单位的时间段，在此期间，Amazon EC2 Auto Scaling 等待实例准备好接收流量，然后继续替换下一个实例。如果您已经正确定义了自动扩缩组的默认实例预热，则无需更改实例预热（除非您想覆盖默认）。有关更多信息，请参阅为 Auto Scaling 组设置原定设置实例预热 (p. 150)。

最低运行正常百分比（**MinHealthyPercentage**）

在可继续刷新之前必须通过组运行状况检查的自动扩缩组所需容量的百分比。有关这些运行状况检查的详细信息，请参阅 Auto Scaling 实例的运行状况检查 (p. 241)。

横向缩减保护实例（**ScaleInProtectedInstances**）

选择若找到横向缩减保护实例时 Amazon EC2 Auto Scaling 要采取的操作。有关这些实例的更多信息，请参阅 实例横向缩减保护 (p. 230)。

Amazon EC2 Auto Scaling 提供以下选项：

- 替换（`Refresh`）- 替换横向缩减保护实例。
- 忽略（`Ignore`）- 忽略横向缩减保护实例，并继续替换未受保护的实例。
- 等待（`Wait`）- 等待一小时，以删除横向缩减保护。如果您不这样做，实例刷新将失败。

跳过匹配（**SkipMatching**）

选择 Amazon EC2 Auto Scaling 是否会跳过替换与所需配置匹配的实例。如果未指定所需配置，则会跳过替换相关实例，这些实例的启动模板和实例类型与自动扩缩组在实例刷新启动之前使用的启动模板和实例类型相同。有关更多信息，请参阅使用实例刷新和跳过匹配 (p. 107)。

备用实例（**StandbyInstances**）

选择 Amazon EC2 Auto Scaling 会在实例处于 `Standby` 状态时做什么。有关这些实例的更多信息，请参阅 临时从 Auto Scaling 组中移除实例 (p. 232)。

Amazon EC2 Auto Scaling 提供以下选项：

- 终止（`Terminate`）- 终止 `Standby` 中的实例。
- 忽略（`Ignore`）- 忽略 `Standby` 中的实例，并继续替换处于 `InService` 状态的实例。
- 等待（`Wait`）- 等待一小时让实例恢复服务。如果您不这样做，实例刷新将失败。

默认值取决于您是从命令行（Amazon CLI 或 Amazon 开发工具包）还是从控制台启动实例刷新。

下表列出了实例刷新设置的默认值。

设置	Amazon CLI 或 Amazon 开发工具包	Amazon EC2 Auto Scaling 控制台
自动回滚	已禁用（`false`）	已禁用
检查点	已禁用（`false`）	已禁用
检查点延迟	1 小时（`3600` 秒）	1 小时
实例预热	默认实例预热 (p. 150)（如果已定义），否则为运行状况检查宽限期 (p. 246)。	默认实例预热 (p. 150)（如果已定义），否则为运行状况检查宽限期 (p. 246)。

设置	Amazon CLI 或 Amazon 开发工具包	Amazon EC2 Auto Scaling 控制台
最低运行正常百分比	90	90
横向缩减保护实例	Wait	Ignore
跳过匹配	已禁用（`false`）	Enabled
备用实例	Wait	Ignore

检查实例刷新的状态

实例刷新开始后，您可以使用 Amazon Web Services Management Console 或 Amazon CLI 获取其状态。

> Tip
>
> 在以下过程中，您需要查看 Auto Scaling 组的 Instance refresh history（实例刷新历史记录）、Activity history（活动历史记录）和 Instances（实例）部分。每个部分应会已经显示已命名的列。要显示隐藏的列或更改显示的行数，请选择每个部分右上角的齿轮图标以打开首选项模式。根据需要更新设置，然后选择确认。

检查实例刷新的状态（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Instance refresh（实例刷新）选项卡上的 Instance refresh history（实例刷新历史记录）下，您可以通过查看 Status（状态）列来确定您的请求的状态。操作在初始化时进入 `Pending` 状态。然后，状态应快速更改为 `InProgress`。所有实例更新后，状态将更改为 `Successful`。
4. 在活动选项卡上的活动历史记录下，当实例刷新开始时，您会看到实例终止时的条目以及启动实例时的另一组条目。在描述列中，您可以找到实例 ID。
5. （可选）如果您有许多扩展活动，则可以通过选择活动历史记录顶部的 **>** 图标来查看其中的更多活动。
6. 在 Instance management (实例管理) 选项卡上的 Instances (实例) 下，可以验证您的实例是否已成功启动。最初，您的实例处于 `Pending` 状态。在实例准备好接收流量时，其状态为 `InService`。运行状况列显示对您的实例执行运行状况检查的结果。

检查实例刷新的状态 (Amazon CLI)

通过使用以下 describe-instance-refreshes 命令，查看 Auto Scaling 组的实例刷新。

```
aws autoscaling describe-instance-refreshes --auto-scaling-group-name my-asg
```

输出示例：

```
{
    "InstanceRefreshes": [
        {
            "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b",
            "AutoScalingGroupName": "my-asg",
            "Status": "InProgress",
            "StartTime": "2020-06-02T18:11:27Z",
            "PercentageComplete": 0,
            "InstancesToUpdate": 5
```

```
        },
        {
            "InstanceRefreshId": "dd7728d0-5bc4-4575-96a3-1b2c52bf8bb1",
            "AutoScalingGroupName": "my-asg",
            "Status": "Successful",
            "StartTime": "2020-06-02T16:43:19Z",
            "EndTime": "2020-06-02T16:53:37Z",
            "PercentageComplete": 100,
            "InstancesToUpdate": 0
        }
    ]
}
```

实例刷新状态

当您启动实例刷新时，它会进入待处理状态。它会从待处理变为正在进行中，直到变为成功、失败、已取消、回滚成功或回滚失败。

实例刷新可以具有以下状态：

状态	描述
Pending	请求已创建，但实例刷新尚未开始。
InProgress	实例刷新正在进行中。
成功	实例刷新已成功完成。
已失败	实例刷新未能完成。您可以使用状态原因和扩展活动进行故障排除。
正在取消	正在取消进行中的实例刷新。
已取消	实例刷新已取消。
正在回滚	正在回滚实例刷新。
回滚失败	回滚未能完成。您可以使用状态原因和扩展活动进行故障排除。
回滚成功	回滚已成功完成。

取消实例刷新

您可以取消仍在进行的实例刷新。在完成该操作后，您无法取消。

取消实例刷新不会回滚任何已替换的实例。要回滚对您的实例所做的更改，请改为执行回滚。有关更多信息，请参阅通过回滚撤消更改 (p. 105)。

主题

- 取消实例刷新（控制台）(p. 104)
- 取消实例刷新（Amazon CLI）(p. 105)

取消实例刷新（控制台）

取消实例刷新

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。

2. 选中 Auto Scaling 组旁边的复选框。
3. 在实例刷新选项卡的活跃实例刷新中，选择操作、取消。
4. 当系统提示进行确认时，选择 Confirm。

实例刷新状态设置为正在取消。取消完成后，实例刷新的状态设置为已取消。

取消实例刷新（Amazon CLI）

取消实例刷新

使用来自 Amazon CLI 的 cancel-instance-refresh 命令，并提供自动扩缩组名。

```
aws autoscaling cancel-instance-refresh --auto-scaling-group-name my-asg
```

输出示例：

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

通过回滚撤消更改

您可以回滚仍在进行的实例刷新。完成后，您无法将其回滚。但是，您可以通过开始新的实例刷新来再次更新您的自动扩缩组。

回滚时，Amazon EC2 Auto Scaling 会替换到目前为止已部署的实例。新实例与您在启动实例刷新之前保存在自动扩缩组中的配置相匹配。

Amazon EC2 Auto Scaling 提供了以下回滚方式：

- 手动回滚：手动开始回滚，以撤消部署到回滚点的内容。
- 自动回滚：如果由于某种原因实例刷新失败，Amazon EC2 Auto Scaling 会自动撤消已部署的内容。

主题

- 先决条件 (p. 105)
- 手动开始回滚（控制台）(p. 106)
- 手动开始回滚（Amazon CLI）(p. 106)
- 使用自动回滚启动实例刷新 (p. 106)

先决条件

要使用回滚功能，您必须执行以下操作：

- 指定所需的配置。
- 确保您上次保存在自动扩缩组中的配置处于稳定状态。如果未处于稳定状态，回滚工作流程仍会发生，但最终会失败。在您解决问题之前，自动扩缩组可能处于失败状态，无法再成功启动实例。这可能会影响服务或应用程序的可用性。
- 确保可以回滚到自动扩缩组的当前启动模板。如果自动扩缩组的启动模板或启动模板版本不兼容，则在启用自动回滚的情况下开始手动回滚或实例刷新时会收到错误。不兼容的启动模板使用 Amazon Systems Manager Parameter Store 中的 AMI 别名。不兼容的启动模板版本包括 `$Latest` 或 `$Default`。

手动开始回滚（控制台）

手动开始实例刷新回滚

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。
3. 在实例刷新选项卡的活跃实例刷新中，依次选择操作、开始回滚。
4. 当系统提示进行确认时，选择 Confirm。

手动开始回滚（Amazon CLI）

手动开始实例刷新回滚

使用来自 Amazon CLI 的 rollback-instance-refresh 命令，并提供自动扩缩组名。

```
aws autoscaling rollback-instance-refresh --auto-scaling-group-name my-asg
```

输出示例：

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

> **Tip**
>
> 如果此命令引发错误，请确保您已在本地将 Amazon CLI 更新到最新版本。

使用自动回滚启动实例刷新

默认情况下，如果实例刷新失败，则不会回滚其应用到新实例的更改。您可以更改行为，以便在实例刷新失败时进行回滚。

本节包含使用自动回滚启动实例刷新的 Amazon CLI 说明。有关如何使用控制台的说明，请参阅启动实例刷新（控制台）(p. 97)。

使用自动回滚启动实例刷新（Amazon CLI）

使用 start-instance-refresh 命令，并将首选项中的自动回滚选项设置为 `true`。此外，提供自动扩缩组名称和所需的配置。

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

`config.json` 的内容。

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredConfiguration": {
      "LaunchTemplate": {
          "LaunchTemplateId": "lt-068f72b729example",
          "Version": "$Default"
       }
    },
    "Preferences": {
      "AutoRollback": true
    }
}
```

如果成功，该命令将返回类似于以下内容的输出。

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Tip

如果此命令引发错误，请确保您已在本地将 Amazon CLI 更新到最新版本。

使用实例刷新和跳过匹配

跳过匹配将通知 Amazon EC2 Auto Scaling 忽略已有最新更新的实例。这样，您就无需替换超过所需数量的实例。当您希望确保自动扩缩组使用特定版本的启动模板，并且只替换那些使用其他版本的实例时，这是很有用的。

在跳过匹配时，请注意以下事项：

- 如果您在启动实例刷新时同时使用跳过匹配和所需配置，Amazon EC2 Auto Scaling 将检查是否有任何实例与所需配置匹配。然后，它只会替换与所需配置不匹配的实例。在实例刷新成功后，Amazon EC2 Auto Scaling 会更新组以反映您的所需配置。
- 如果您使用跳过匹配启动实例刷新，但没有指定所需的配置，Amazon EC2 Auto Scaling 会检查是否有任何实例与您上次保存在自动扩缩组中的配置相匹配。然后，它只会替换与您上次保存的配置不匹配的实例。
- 您可以将跳过匹配与新的启动模板、启动模板的新版本或一组实例类型配合使用。如果您启用了跳过匹配，但这些内容都没有更改，则实例刷新将立即成功，而无需替换任何实例。如果您对所需配置进行了任何其他更改（例如更改竞价型分配策略），Amazon EC2 Auto Scaling 将等待实例刷新成功。然后，它会更新自动扩缩组设置以反映新的所需配置。
- 您不能将跳过匹配与新的启动配置配合使用。
- 在使用跳过匹配进行实例刷新期间，如果为启动模板指定了 `$Default` 或 `$Latest`，Amazon EC2 Auto Scaling 会在替换每个实例时重新评估启动模板版本。因此，如果您在实例仍在替换时创建启动模板的新版本，则 Amazon EC2 Auto Scaling 最终可能会使用启动模板的新版本。当 Amazon EC2 Auto Scaling 在整个组中应用新版本时，这可能会导致先前替换的实例再次被替换。

本节包含在启用跳过匹配的情况下启动实例刷新的 Amazon CLI 说明。有关如何使用控制台的说明，请参阅启动实例刷新（控制台）(p. 97)。

跳过匹配（基本程序）

请按照本部分中的步骤，使用 Amazon CLI 来执行以下操作：

- 创建您希望应用于实例的启动模板。
- 启动实例刷新，将启动模板应用到自动扩缩组。如果您不启用跳过匹配，则所有实例都将被替换。即使用于预置实例的启动模板与您为所需配置指定的启动模板相同，也是如此。

将跳过匹配与新的启动模板配合使用

1. 使用 create-launch-template 命令为自动扩缩组创建新的启动模板。包括定义为您的自动扩缩组创建的实例详细信息的 `--launch-template-data` 选项和 JSON 输入。

   例如，使用以下命令创建包含 AMI ID *ami-0123456789abcdef0* 和 *t2.micro* 实例类型的基本启动模板。

   ```
   aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
   version-description version1 \
   ```

```
  --launch-template-data
 '{"ImageId":"ami-0123456789abcdef0","InstanceType":"t2.micro"}'
```

如果成功，该命令将返回类似于以下内容的输出。

```
{
   "LaunchTemplate": {
     "LaunchTemplateId": "lt-068f72b729example",
     "LaunchTemplateName": "my-template-for-auto-scaling",
     "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
     "CreateTime": "2023-01-30T18:16:06.000Z",
     "DefaultVersionNumber": 1,
     "LatestVersionNumber": 1
  }
}
```

有关更多信息，请参阅使用 Amazon Command Line Interface (Amazon CLI) 创建和管理启动模板的示例 (p. 29)。

2. 使用 start-instance-refresh 命令启动实例替换工作流程，并应用包含 ID *lt-068f72b729example* 的新启动模板。因为启动模板是新的，所以它只有一个版本。这意味着启动模板的版本 1 是本次实例刷新的目标。如果在实例刷新期间发生横向扩展事件，并且 Amazon EC2 Auto Scaling 使用本启动模板的版本 1 预置新的实例，则它们不会被替换。成功完成操作后，新的启动模板即成功应用到自动扩缩组。

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

config.json 的内容。

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredConfiguration": {
      "LaunchTemplate": {
          "LaunchTemplateId": "lt-068f72b729example",
          "Version": "$Default"
       }
    },
    "Preferences": {
      "SkipMatching": true
    }
}
```

如果成功，该命令将返回类似于以下内容的输出。

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

跳过匹配（混合实例组）

如果您的自动扩缩组采用混合实例策略 (p. 50)，请按照本节中的步骤在使用跳过匹配的情况下利用 Amazon CLI 启动实例刷新。您有以下选项：

- 提供新的启动模板，以应用于在策略中指定的所有实例类型。
- 提供一组更新的实例类型，无论是否更改策略中的启动模板。例如，您可能想要从不需要的实例类型迁移。您可以按原样使用启动模板，无需更改 AMI、安全组或要替换实例的其他具体信息。

请按照以下部分之一的步骤操作，具体取决于哪个选项适合您的需求。

将跳过匹配与新的启动模板配合使用

1. 使用 create-launch-template 命令为自动扩缩组创建新的启动模板。包括定义为您的自动扩缩组创建的实例详细信息的 `--launch-template-data` 选项和 JSON 输入。

   例如，使用以下命令创建包含 AMI ID *ami-0123456789abcdef0* 的启动模板。

   ```
   aws ec2 create-launch-template --launch-template-name my-new-template --version-
   description version1 \
     --launch-template-data '{"ImageId":"ami-0123456789abcdef0"}'
   ```

   如果成功，该命令将返回类似于以下内容的输出。

   ```
   {
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-04d5cc9b88example",
        "LaunchTemplateName": "my-new-template",
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2023-01-31T15:56:02.000Z",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
     }
   }
   ```

   有关更多信息，请参阅使用 Amazon Command Line Interface (Amazon CLI) 创建和管理启动模板的示例 (p. 29)。

2. 要查看自动扩缩组的现有混合实例策略，请运行 describe-auto-scaling-groups 命令。在下一步中，当您开始刷新实例时，您将需要这些信息。

   以下示例命令返回为名为 *my-asg* 的自动扩缩组配置的混合实例策略。

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
   ```

   如果成功，该命令将返回类似于以下内容的输出。

   ```
   {
     "AutoScalingGroups":[
       {
         "AutoScalingGroupName":"my-asg",
         "AutoScalingGroupARN":"arn",
         "MixedInstancesPolicy":{
           "LaunchTemplate":{
             "LaunchTemplateSpecification":{
               "LaunchTemplateId":"lt-073693ed27example",
               "LaunchTemplateName":"my-old-template",
               "Version":"$Default"
             },
             "Overrides":[
               {
                 "InstanceType":"c5.large"
               },
               {
                 "InstanceType":"c5a.large"
               },
               {
                 "InstanceType":"m5.large"
               },
               {
                 "InstanceType":"m5a.large"
               }
   ```

```
          ]
        },
        "InstancesDistribution":{
          "OnDemandAllocationStrategy":"prioritized",
          "OnDemandBaseCapacity":1,
          "OnDemandPercentageAboveBaseCapacity":50,
          "SpotAllocationStrategy":"price-capacity-optimized"
        }
      },
      "MinSize":1,
      "MaxSize":5,
      "DesiredCapacity":4,
      ...
    }
  ]
}
```

3. 使用 start-instance-refresh 命令启动实例替换工作流程，并应用包含 ID *lt-04d5cc9b88example* 的新启动模板。因为启动模板是新的，所以它只有一个版本。这意味着启动模板的版本 1 是本次实例刷新的目标。如果在实例刷新期间发生横向扩展事件，并且 Amazon EC2 Auto Scaling 使用本启动模板的版本 1 预置新的实例，则它们不会被替换。成功完成操作后，更新的混合实例策略将成功应用到自动扩缩组。

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

config.json 的内容。

```
{
  "AutoScalingGroupName":"my-asg",
  "DesiredConfiguration":{
    "MixedInstancesPolicy":{
      "LaunchTemplate":{
        "LaunchTemplateSpecification":{
          "LaunchTemplateId":"lt-04d5cc9b88example",
          "Version":"$Default"
        },
        "Overrides":[
          {
            "InstanceType":"c5.large"
          },
          {
            "InstanceType":"c5a.large"
          },
          {
            "InstanceType":"m5.large"
          },
          {
            "InstanceType":"m5a.large"
          }
        ]
      },
      "InstancesDistribution":{
        "OnDemandAllocationStrategy":"prioritized",
        "OnDemandBaseCapacity":1,
        "OnDemandPercentageAboveBaseCapacity":50,
        "SpotAllocationStrategy":"price-capacity-optimized"
        }
      }
    }
  },
  "Preferences":{
    "SkipMatching":true
  }
```

```
}
```

如果成功，该命令将返回类似于以下内容的输出。

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

在下一个步骤中，您无需更改启动模板即可提供一组更新的实例类型。

将跳过匹配与一组更新的实例类型配合使用

1. 要查看自动扩缩组的现有混合实例策略，请运行 describe-auto-scaling-groups 命令。在下一步中，当您开始刷新实例时，您将需要这些信息。

   以下示例命令返回为名为 *my-asg* 的自动扩缩组配置的混合实例策略。

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
   ```

   如果成功，该命令将返回类似于以下内容的输出。

   ```
   {
     "AutoScalingGroups":[
       {
         "AutoScalingGroupName":"my-asg",
         "AutoScalingGroupARN":"arn",
         "MixedInstancesPolicy":{
           "LaunchTemplate":{
             "LaunchTemplateSpecification":{
               "LaunchTemplateId":"lt-073693ed27example",
               "LaunchTemplateName":"my-template-for-auto-scaling",
               "Version":"$Default"
             },
             "Overrides":[
               {
                 "InstanceType":"c5.large"
               },
               {
                 "InstanceType":"c5a.large"
               },
               {
                 "InstanceType":"m5.large"
               },
               {
                 "InstanceType":"m5a.large"
               }
             ]
           },
           "InstancesDistribution":{
             "OnDemandAllocationStrategy":"prioritized",
             "OnDemandBaseCapacity":1,
             "OnDemandPercentageAboveBaseCapacity":50,
             "SpotAllocationStrategy":"price-capacity-optimized"
           }
         },
         "MinSize":1,
         "MaxSize":5,
         "DesiredCapacity":4,
         ...
       }
   ```

```
  ]
}
```

2. 使用 start-instance-refresh 命令启动实例替换工作流程并应用更新。如果您希望替换使用特定实例类型的实例，您所需的配置必须指定仅含所需实例类型的混合实例策略。您可以选择是否在这些实例类型上添加新的实例类型。

   以下示例命令在没有不需要的实例类型 *m5a.large* 的情况下启动实例刷新。当组中的实例类型与其余三个实例类型之一不匹配时，实例将被替换。（请注意，实例刷新不会选择要从中预置新实例的实例类型；而是由分配策略 (p. 50)来选择。）成功完成操作后，更新的混合实例策略将成功应用到自动扩缩组。

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

   config.json 的内容

```
{
  "AutoScalingGroupName":"my-asg",
  "DesiredConfiguration":{
    "MixedInstancesPolicy":{
      "LaunchTemplate":{
        "LaunchTemplateSpecification":{
          "LaunchTemplateId":"lt-073693ed27example",
          "Version":"$Default"
        },
        "Overrides":[
          {
            "InstanceType":"c5.large"
          },
          {
            "InstanceType":"c5a.large"
          },
          {
            "InstanceType":"m5.large"
          }
        ]
      },
      "InstancesDistribution":{
        "OnDemandAllocationStrategy":"prioritized",
        "OnDemandBaseCapacity":1,
        "OnDemandPercentageAboveBaseCapacity":50,
        "SpotAllocationStrategy":"price-capacity-optimized"
        }
      }
    }
  },
  "Preferences":{
    "SkipMatching":true
  }
}
```

将检查点添加到实例刷新

在使用实例刷新时，您可以选择分阶段替换实例，以便您可以随时对实例执行验证。要执行分阶段替换，请添加检查点，这些检查点是实例刷新暂停时的时间点。使用检查点使您能够更好地控制选择更新 Auto Scaling 组的方式。它可以帮助您确认您的应用程序将以可靠、可预测的方式运行。

Amazon EC2 Auto Scaling 会针对每个检查点发出事件。您可以添加 EventBridge 规则，将事件发送到目标（如 Amazon SNS）。这样，当您可以运行所需的验证时，系统就会通知您。有关更多信息，请参阅为实例刷新事件创建 EventBridge 规则 (p. 297)。

目录

- 注意事项 (p. 113)
- 启用检查点（控制台） (p. 113)
- 启用检查点 (Amazon CLI) (p. 114)

注意事项

在使用检查点时，请记住以下注意事项：

- 当替换的实例数达到为检查点定义的百分比阈值时，将达到检查点。替换的实例所占的百分比可以等于或高于百分比阈值，但不能低于该阈值。
- 达到检查点后，总体完成百分比不会显示最新状态，直到实例完成预热后才会显示。

  例如，假设您的 Auto Scaling 组有 10 个实例。您的检查点百分比为 `[20,50]`，其中具有 15 分钟的检查点延迟和 80 % 的最低运行正常百分比。您的组将进行以下替换：
  - 0:00：将两个旧实例替换为新实例。
  - 0:10：两个新实例完成预热。
  - 0:25：将两个旧实例替换为新实例。（为了保持最低运行正常百分比，仅替换两个实例。）
  - 0:35：两个新实例完成预热。
  - 0:35：将一个旧实例替换为新实例。
  - 0:45：一个新实例完成预热。

  在 0:35 时，操作将停止启动新实例。完成百分比尚不能准确反映已完成替换的数量 (50%)，因为新实例未完成预热。在新实例于 0:45 完成其预热期后，完成百分比将显示 50%。
- 由于检查点基于百分比，因此要替换的实例数将随组的大小而变化。当发生横向扩展活动并且组的大小增加时，正在进行的操作可能会再次达到检查点。如果发生这种情况，Amazon EC2 Auto Scaling 会发送另一个通知，并在继续之前重复检查点之间的等待时间。
- 在某些情况下可以跳过检查点。例如，假设您的 Auto Scaling 组有两个实例，并且您的检查点百分比为 `[10,40,100]`。替换第一个实例后，Amazon EC2 Auto Scaling 将计算出已替换该组的 50%。由于 50% 高于前两个检查点，因此它将跳过第一个检查点 (`10`) 并发送第二个检查点 (`40`) 的通知。
- 取消操作将停止进行任何进一步替换。如果您取消操作或在达到最后一个检查点之前操作失败，则任何已替换的实例都不会回滚到其以前的配置。
- 对于部分刷新，当您重新运行操作时，Amazon EC2 Auto Scaling 不会从最后一个检查点的时刻重新开启，也不会在仅替换旧实例时停止。但是，它将首先针对旧实例进行替换，然后再针对新实例。

启用检查点（控制台）

您可以在开启实例刷新之前启用检查点，以使用增量或分阶段方法替换实例。这将为验证提供额外时间。

启动使用检查点的实例刷新

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling group（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Instance refresh（实例刷新）选项卡上的 Active instance refresh（活跃实例刷新）中，选择 Start instance refresh（开启实例刷新）。
4. 在 Start instance refresh（开启实例刷新）页面上，输入 Minimum healthy percentage（最低运行正常百分比）和 Instance warmup（实例预热）的值。
5. 选择 Enable checkpoints（启用检查点）复选框。

此时将显示一个框，您可以在其中定义第一个检查点的百分比阈值。

6. 对于 Proceed until ____ % of the group is refreshed（继续直到刷新该组的 ____ %）中，输入一个数字 (1–100)。这将设置第一个检查点的百分比。
7. 要添加另一个检查点，请选择添加检查点，然后定义下一个检查点的百分比。
8. 要指定 Amazon EC2 Auto Scaling 在达到检查点后等待多长时间，请更新在检查点之间等待 **1 hour** 中的字段。时间单位可以是小时、分钟或秒。
9. 如果您已完成实例刷新选择，请选择 Start（开启）。

启用检查点 (Amazon CLI)

要在已使用 Amazon CLI 启用检查点的情况下开启实例刷新，您需要定义以下参数的配置文件：

- `CheckpointPercentages`：指定要替换的实例百分比的阈值。这些阈值提供检查点。当替换和预热的实例百分比达到指定阈值之一时，操作将等待指定的时间段。您可指定在 `CheckpointDelay` 中等待的秒数。当指定的时间段过后，实例刷新将继续进行直到达到下一个检查点（如果适用）。
- `CheckpointDelay`：指定在到达检查点之后与继续操作之前需要等待的时间量（以秒为单位）。选择提供足够时间以执行验证的时间段。

`CheckpointPercentages` 数组中显示的最后一个值描述需要成功替换的 Auto Scaling 组的百分比。在成功替换此百分比，并且每个实例都已预热且准备好再次提供流量后，该操作将转换到 `Successful`。

创建多个检查点

要创建多个检查点，请使用以下示例 start-instance-refresh 命令。此示例配置了一个实例刷新，它最初刷新 Auto Scaling 组的 1%。在等待 10 分钟后，它会随后刷新接下来的 19%，然后再等待 10 分钟。最后，它会刷新该组的其余部分，然后结束操作。

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

`config.json` 的内容：

```
{
    "AutoScalingGroupName": "my-asg",
    "Preferences": {
      "InstanceWarmup": 60,
      "MinHealthyPercentage": 80,
      "CheckpointPercentages": [1,20,100],
      "CheckpointDelay": 600
    }
}
```

创建单个检查点

要创建单个检查点，请使用以下示例 start-instance-refresh 命令。此示例配置了一个实例刷新，它最初刷新 Auto Scaling 组的 20%。在等待 10 分钟后，它会随后刷新该组的其余部分，然后结束操作。

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

`config.json` 的内容：

```
{
    "AutoScalingGroupName": "my-asg",
    "Preferences": {
      "InstanceWarmup": 60,
```

```
      "MinHealthyPercentage": 80,
      "CheckpointPercentages": [20,100],
      "CheckpointDelay": 600
    }
}
```

部分刷新 Auto Scaling 组

要仅替换 Auto Scaling 组的一部分，然后完全停止，请使用以下示例 start-instance-refresh 命令。此示例配置了一个实例刷新，它最初刷新 Auto Scaling 组的 1%。在等待 10 分钟后，它会随后刷新接下来的 19%，然后结束操作。

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

config.json 的内容：

```
{
    "AutoScalingGroupName": "my-asg",
    "Preferences": {
      "InstanceWarmup": 60,
      "MinHealthyPercentage": 80,
      "CheckpointPercentages": [1,20],
      "CheckpointDelay": 600
    }
}
```

基于最大实例生命周期替换 Auto Scaling 实例

最大实例生命周期指定实例在终止和替换之前可提供服务的最长时间（以秒为单位）。由于内部安全策略或外部合规性控制，常见使用案例可能需要按计划替换您的实例。

您必须指定至少 86,400 秒（一天）的值。要清除以前设置的值，请指定新值 0。此设置适用于 Auto Scaling 组中的所有当前和未来实例。

将此值设置得太低可能会导致实例的替换速度超出预期。一般来说，Amazon EC2 Auto Scaling 一次替换一个实例，两次替换之间会有暂停。但是，如果您指定的最大实例生命周期无法足够时间来单独替换每个实例，Amazon EC2 Auto Scaling 必须一次替换多个实例。可能会一次替换多个实例，最多可达 Auto Scaling 组当前容量的 10%。

要管理替换率，您可以执行以下操作：

- 应将最大实例生命周期限制设置为较长的时间段。这会将替换分隔开，对于有大量实例要替换的组很有帮助。
- 可以使用实例保护在特定替换之间添加额外时间。这将暂时防止替换 Auto Scaling 组中的单个实例。当您准备好替换这些实例时，请从每个单独的实例中删除实例保护。有关更多信息，请参阅实例横向缩减保护 (p. 230)。

  Note

  每当替换旧实例并启动新实例时，新实例都会使用当前与自动扩缩组关联的启动模板或启动配置。如果启动模板或启动配置指定了其他版本应用程序的 AMI ID，则将自动部署此版本的应用程序。

设置最大实例生命周期

在控制台中创建 Auto Scaling 组时，您将无法设置最大实例生命周期。但在创建该组后，您就可以对其进行编辑，以设置最大实例生命周期。

为组设置最大实例生命周期（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格，其中显示有关您选择的组的信息。
3. 在 Details（详细信息）选项卡上，选择 Advanced configurations（高级配置）、Edit（编辑）。
4. 对于 Maximum instance lifetime (最大实例生命周期)，输入实例可使用的最长秒数。
5. 选择 Update（更新）。

在 Activity（活动）选项卡上的 Activity history（活动历史记录）下，您可以通过其整个历史记录查看该组中实例的替换情况。

为组设置最大实例生命周期 (Amazon CLI)

您也可以使用 Amazon CLI 为新的或现有 Auto Scaling 组设置最大实例生命周期。

对于新的 Auto Scaling 组，请使用 create-auto-scaling-group 命令。

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

以下示例 config.json 文件显示的最长实例生命周期为 2592000 秒（30 天）。

```
{
    "AutoScalingGroupName": "my-asg",
    "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Latest"
    },
    "MinSize": 1,
    "MaxSize": 5,
    "MaxInstanceLifetime": 2592000,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "Tags": []
}
```

对于现有的 Auto Scaling 组，请使用 update-auto-scaling-group 命令。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-existing-asg --max-
instance-lifetime 2592000
```

验证 Auto Scaling 组的最大实例生命周期

使用 describe-auto-scaling-groups 命令。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

以下为响应示例。

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            "LaunchTemplate": {
```

```
                "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
                "LaunchTemplateName": "my-launch-template",
                "Version": "$Latest"
            },
            "MinSize": 1,
            "MaxSize": 5,
            "DesiredCapacity": 1,
            "DefaultCooldown": 300,
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2b",
                "us-west-2c"
            ],
            "LoadBalancerNames": [],
            "TargetGroupARNs": [],
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": 0,
            "Instances": [
                {
                    "InstanceId": "i-04d180b9d5fc578fc",
                    "InstanceType": "t2.small",
                    "AvailabilityZone": "us-west-2b",
                    "LifecycleState": "Pending",
                    "HealthStatus": "Healthy",
                    "LaunchTemplate": {
                        "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "7"
                    },
                    "ProtectedFromScaleIn": false
                }
            ],
            "CreatedTime": "2019-11-14T22:56:15.487Z",
            "SuspendedProcesses": [],
            "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
            "EnabledMetrics": [],
            "Tags": [],
            "TerminationPolicies": [
                "Default"
            ],
            "NewInstancesProtectedFromScaleIn": false,
            "ServiceLinkedRoleARN": "arn",
            "MaxInstanceLifetime": 2592000
        }
    ]
}
```


限制

- 不能保证每个实例的最大生命周期都是准确的：不能保证仅在实例最大持续时间结束时替换实例。在某些情况下，Amazon EC2 Auto Scaling 可能需要在您更新最大实例生命周期参数后立即开始替换实例。此行为的原因是避免同时替换所有实例。
- 实例在启动前终止：当 Auto Scaling 组中只有一个实例时，最大实例生命周期功能可能会导致中断，因为 Amazon EC2 Auto Scaling 会终止实例，然后启动新实例。

删除 Auto Scaling 基础设施

要完全删除您的扩展基础设施，请完成以下任务。

任务

- 删除 Auto Scaling 组 (p. 118)
- （可选）删除启动配置 (p. 118)
- （可选）删除启动模板 (p. 119)
- （可选）删除负载均衡器和目标组 (p. 119)
- （可选）删除 CloudWatch 警报 (p. 120)

删除 Auto Scaling 组

当您删除 Auto Scaling 组时，其所需值、最小值和最大值设置为 0。因此，将会终止实例。删除实例还会删除任何关联的日志或数据，以及该实例上的任何卷。如果不想终止一个或多个实例，您可在删除 Auto Scaling 组之前分离它们。如果组具有扩展策略，则在删除组时，将会删除策略、基础警报操作以及不再具有关联操作的任何警报。

删除 Auto Scaling 组（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框并选择 Delete（删除）。
3. 当系统提示进行确认时，键入 **delete** 以确认删除指定自动扩缩组，然后选择 Delete（删除）。

   Name（名称）列中的加载图标指示 Auto Scaling 组正在被删除。Desired（所需）、Min（最小）和 Max（最大）列显示 Auto Scaling 组具有 0 个实例。终止实例并删除组需要几分钟时间。刷新列表以查看当前状态。

要删除 Auto Scaling 组 (Amazon CLI)

使用以下 delete-auto-scaling-group 命令删除 Auto Scaling 组。如果该组有任何 EC2 实例，则此操作不起作用；它仅适用于具有零个实例的组。

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name my-asg
```

如果该组具有正在进行的实例或扩展活动，请使用包含 `--force-delete` 选项的 delete-auto-scaling-group 命令。这同时将终止 EC2 实例。当您从 Amazon EC2 Auto Scaling 控制台中删除自动扩缩组时，控制台使用此操作终止所有 EC2 实例并同时删除该组。

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name my-asg --force-delete
```

（可选）删除启动配置

要保留启动配置以备将来使用，可跳过此步骤。

删除启动配置（控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的 Auto Scaling (自动调整) 上，选择 Launch Configurations (启动配置)。
3. 在 Launch configurations（启动配置）页面上，选择启动配置，然后依次选择 Actions（操作）和 Delete launch configuration（删除启动配置）。
4. 当系统提示进行确认时，选择 Delete（删除）。

删除启动配置 (Amazon CLI)

使用以下 delete-launch-configuration 命令。

```
aws autoscaling delete-launch-configuration --launch-configuration-name my-launch-config
```

（可选）删除启动模板

您可以删除启动模板或仅删除启动模板的某个版本。在删除启动模板时，将删除其所有版本。

您可以跳过此步骤来保留启动模板以供将来使用。

删除启动模板（控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的实例下，选择启动模板。
3. 选择启动模板，然后执行下列操作之一：
   - 选择 Actions（操作），然后选择 Delete template（删除模板）。当系统提示进行确认时，键入 **Delete** 以确认删除指定启动模板，然后选择 Delete（删除）。
   - 选择 Actions（操作），然后选择 Delete template version（删除模板版本）。选择要删除的版本，然后选择 Delete (删除)。

删除启动模板 (Amazon CLI)

使用以下 delete-launch-template 命令可删除您的模板及其所有版本。

```
aws ec2 delete-launch-template --launch-template-id lt-068f72b72934aff71
```

或者，您也可以使用 delete-launch-template-versions 命令删除启动模板的特定版本。

```
aws ec2 delete-launch-template-versions --launch-template-id lt-068f72b72934aff71 --
versions 1
```

（可选）删除负载均衡器和目标组

如果未在 Elastic Load Balancing 负载均衡器中关联 Auto Scaling 组，或者要保留负载均衡器以供将来使用，请跳过该步骤。

删除您的负载均衡器（控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
3. 选择负载均衡器，然后依次选择 Actions（操作）和 Delete（删除）。
4. 当系统提示进行确认时，选择 Yes, Delete（是，删除）。

删除目标组（控制台）

1. 在导航窗格上的 Load Balancing (负载均衡) 下，选择 Target Groups (目标组)。
2. 选择目标组，然后依次选择 Actions（操作）、Delete（删除）。
3. 当系统提示进行确认时，选择 Yes, Delete（是，删除）。

删除与 Auto Scaling 组关联的负载均衡器 (Amazon CLI)

对于 Application Load Balancer 和 Network Load Balancer，请使用以下 delete-load-balancer 和 delete-target-group 命令。

```
aws elbv2 delete-load-balancer --load-balancer-arn my-load-balancer-arn
aws elbv2 delete-target-group --target-group-arn my-target-group-arn
```

要删除您的经典负载均衡器，请使用以下 delete-load-balancer 命令。

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

（可选）删除 CloudWatch 警报

要删除与 Auto Scaling 组关联的任何 CloudWatch 警报，请完成以下步骤。

如果 Auto Scaling 组未与任何 CloudWatch 警报关联，或者您要保留警报以备将来使用，可跳过此步骤。

> **Note**
>
> 删除 Auto Scaling 组会自动删除 Amazon EC2 Auto Scaling 为目标跟踪扩展策略管理的 CloudWatch 警报。

删除 CloudWatch 警报（控制台）

1. 通过以下网址打开 CloudWatch 控制台：https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格上，选择 Alarms（警报）。
3. 选择警报，然后选择 Action（操作）、Delete（删除）。
4. 当系统提示进行确认时，选择 Delete（删除）。

删除 CloudWatch 警报 (Amazon CLI)

使用 delete-alarms 命令。您可以一次删除一个或多个警报。例如，使用以下命令可删除 `Step-Scaling-AlarmHigh-AddCapacity` 和 `Step-Scaling-AlarmLow-RemoveCapacity` 警报：

```
aws cloudwatch delete-alarms --alarm-name Step-Scaling-AlarmHigh-AddCapacity Step-Scaling-AlarmLow-RemoveCapacity
```

扩展 Auto Scaling 组的大小

扩展是增加或减少应用程序的计算容量的能力。扩缩从一个事件或扩缩操作开始，以指示 Auto Scaling 组启动或终止 Amazon EC2 实例。

Amazon EC2 Auto Scaling 为满足您的应用程序需求提供了多种调整扩缩的方式。因此，您需要很好地了解您的应用程序，这十分重要。请注意以下事项：

- Amazon EC2 Auto Scaling 应该在您的应用程序架构中发挥什么作用？ 通常可以将自动扩展视为一种增加和减少容量的主要方法，但自动扩展在保持稳定数量的服务器方面也是非常有用的。
- 哪些成本约束对您比较重要？ 由于 Amazon EC2 Auto Scaling 使用 EC2 实例，您只需为使用的资源付费了解成本约束可以帮助您确定何时扩展应用程序以及扩展量。
- 哪些指标对您的应用程序比较重要？ Amazon CloudWatch 支持多种可用于 Auto Scaling 组的不同指标。

目录

- 扩展选项 (p. 121)
- 为 Auto Scaling 组设置容量限制 (p. 122)
- 在 Auto Scaling 组中保持固定数量的实例 (p. 123)
- Amazon EC2 Auto Scaling 的手动扩缩 (p. 123)
- Amazon EC2 Auto Scaling 的动态扩缩 (p. 132)
- Amazon EC2 Auto Scaling 的预测式扩展 (p. 166)
- Amazon EC2 Auto Scaling 的计划扩缩 (p. 186)
- Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)
- Amazon EC2 Auto Scaling 的暖池 (p. 212)
- 控制在横向缩减过程中要终止的 Auto Scaling 实例 (p. 222)
- 临时从 Auto Scaling 组中移除实例 (p. 232)
- 暂停和恢复 Auto Scaling 组的进程 (p. 236)

扩展选项

Amazon EC2 Auto Scaling 提供了多种方法来扩缩您的 Auto Scaling 组。

始终保持当前实例级别

您可以配置 Auto Scaling 组，使其始终保持指定的运行实例数。为了保持当前实例等级，Amazon EC2 Auto Scaling 将对 Auto Scaling 组内运行的实例执行定期运行状况检查。如果 Amazon EC2 Auto Scaling 发现运行状况不佳的实例，它将终止该实例，并启动新实例。有关更多信息，请参阅 在 Auto Scaling 组中保持固定数量的实例 (p. 123)。

手动缩放

手动扩缩是扩缩资源的最基本方法，您只需指定 Auto Scaling 组的最大容量、最小容量或所需容量的变化。Amazon EC2 Auto Scaling 可以管理创建或终止实例的流程来维护更新的容量。有关更多信息，请参阅 Amazon EC2 Auto Scaling 的手动扩缩 (p. 123)。

按计划扩展

按计划扩展意味着扩展操作作为时间和日期的函数自动执行。这在您确切地知道何时增加或减少组中的实例数量时非常有用，因为该需求遵循可预测的计划。有关更多信息，请参阅Amazon EC2 Auto Scaling 的计划扩缩 (p. 186)。

根据需求进行扩展

使用动态扩缩是一种更高级的资源扩缩方法，您可以定义扩缩策略，以动态调整 Auto Scaling 组的大小以满足需求的变化。例如，假设您有一个当前在两个实例上运行的 Web 应用程序，并希望在应用程序负载变化时将 Auto Scaling 组的 CPU 使用率保持在 50% 左右。在根据条件变化进行扩展，但却不知道条件何时改变时，可以使用这种方法。您可以设置 Amazon EC2 Auto Scaling 以进行响应。有关更多信息，请参阅 Amazon EC2 Auto Scaling 的动态扩缩 (p. 132)。

使用预测式扩展

您还可以将预测性扩缩和动态扩缩（分别为主动和被动方法）结合使用，以提高 EC2 容量扩缩的速度。使用预测式扩展可在流量流的每日和每周模式之前增加 Auto Scaling 组中的 EC2 实例数。有关更多信息，请参阅Amazon EC2 Auto Scaling 的预测式扩展 (p. 166)。

为 Auto Scaling 组设置容量限制

容量限制表示所需自动扩缩组的最小组大小和最大组大小。您可以单独设置最小和最大大小的限制。

组的所需容量可以调整为位于最小和最大大小限制范围内的数值。所需容量必须大于等于组的最小大小，小于等于组的最大大小。

- 所需容量：表示自动扩缩组在创建时的初始容量。自动扩缩组会尝试保持所需容量。它首先按照为所需容量指定的数量启动实例，如果自动扩缩组没有附加任扩缩策略或计划操作，则将维持这一实例数量。
- 最小容量：表示最小组大小。设定扩缩策略后，自动扩缩组不能将其所需容量降至最小大小限制以下。
- 最大容量：表示最大组大小。设定扩缩策略后，自动扩缩组不能将其所需容量降至增加至最小大小限制以上。

最小和最大大小限制也适用于以下场景：

- 当您通过更新所需容量来手动扩展自动扩缩组时。
- 会更新所需容量的计划操作运行时。如果计划操作运行时没有为组指定新的最小和最大大小限制，则会使用该组当前的最小和最大大小限制。

自动扩缩组始终会尝试保持维持所需容量。如果某个实例意外终止（例如，由于竞价型实例中断、运行状况检查失败或人工操作），该组会自动启动一个新实例以维持所需容量。

要在控制台中管理这些设置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格的 Auto Scaling 下，选择 自动扩缩组。
3. 在 Auto Scaling groups（自动扩缩组）页面中，选择您的自动扩缩组旁的复选框。

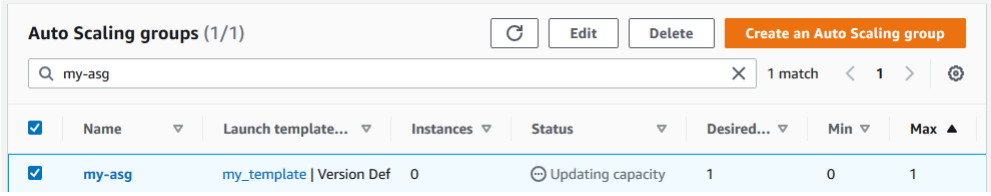
   这时将在页面底部打开一个拆分窗格。
4. 在详细信息选项卡的下窗格，查看或更改最小容量、最大容量和所需容量的当前设置。有关更多信息，请参阅更改 Auto Scaling 组的大小（控制台）(p. 123)。

在 Details（详细信息）窗格上方，您可以找到自动扩缩组中的当前实例数、最小容量、最大容量和所需容量以及状态列等信息。如果自动扩缩组使用实例权重，则还可以找到贡献到所需容量的容量单位数信息。

要在列表中添加或删除列，请选择页面顶部的设置图标。然后，对于 Auto Scaling groups attributes（自动扩缩组属性），打开或关闭每个列，然后选择 Confirm（确认）。

更改后验证 Auto Scaling 组的大小

实例列显示当前正在运行的实例数。启动或终止实例时，状态列显示正在更新容量状态，如下图所示。



请等待几分钟，然后刷新视图以查看最新状态。某个扩缩活动完成后，Instances（实例）列将会显示更新后的值。

您可以在 Instance management（实例管理）选项卡的 Instances（实例）下查看当前正在运行的实例数量和状态。

Note

您可以使用 Service Quotas 更新您的 Amazon Web Services 账户 中的 EC2 实例和其他资源的总容量限制。在 Service Quotas 控制台中，您可以查看所有可用的服务限额并申请增加这些限额。有关更多信息，请参阅《Service Quotas 用户指南》中的申请增加限额。

在 Auto Scaling 组中保持固定数量的实例

Amazon EC2 Auto Scaling 支持设置 Auto Scaling 组以保持固定大小。然后，您可以选择是调整组的所需容量，还是手动添加或移除组中的 Amazon EC2 实例，应对应用程序的流量更改。

如果需要固定数量的实例，则可以为最小容量、最大容量和所需容量设置相同值来实现。创建 Auto Scaling 组后，组先启动足够的 EC2 实例以满足其最低容量。如果 Auto Scaling 组没有附加其他扩展条件，组始终保持这一最小数量的运行实例。

即使某个实例运行状况不佳，您的 Auto Scaling 组也会继续保持固定数量的实例。Amazon EC2 Auto Scaling 监控每个 Auto Scaling 实例的运行状况。如果发现实例运行状况不佳，它将终止该实例，并启动新实例。由于各种可能的原因，实例的运行状况检查可能会失败。有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。

Amazon EC2 Auto Scaling 的手动扩缩

您可以随时手动更改现有 Auto Scaling 组的大小。您可以更新 Auto Scaling 组的所需容量，或更新附加到 Auto Scaling 组的实例。当不需要自动扩展或需要将容量保持为固定数量的实例时，手动扩展组非常有用。

更改 Auto Scaling 组的大小（控制台）

在更改 Auto Scaling 组的所需容量时，Amazon EC2 Auto Scaling 将管理启动或终止实例的过程以维护新的组大小。

以下示例假设您创建了一个最小大小为 1，最大大小为 5 的 Auto Scaling 组。因此，该组目前正在运行的实例是 1 个。

要更改 Auto Scaling 组的大小

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details（详细信息）选项卡上，选择 Group details（组详细信息）、Edit (编辑)。
4. 对于 Desired capacity（所需容量），请将所需容量增加一。例如，如果当前值是 1，请输入 2。

理想容量必须小于或等于组的最大容量。如果 Desired capacity（所需容量）的新值大于 Maximum capacity（最大容量），则必须更新 Maximum capacity（最大容量）。

5. 完成后，选择 Update（更新）。

接下来，验证 Auto Scaling 组是否已另外启动一个实例。

要验证 Auto Scaling 组的容量是否已更改

1. 在 Activity（活动）选项卡上的 Activity history（活动历史状态）中，Status（状态）列显示实例的当前状态。使用刷新按钮，直到您看到实例的状态更改为 Successful (成功)。这表明您的 Auto Scaling 组已成功启动新实例。

   Note

   如果实例无法启动，您可以在Amazon EC2 Auto Scaling 问题排查 (p. 341)中找到问题排查提示。

2. 在 Instance management (实例管理) 选项卡的 Instances (实例) 中，Lifecycle (生命周期) 列显示实例的状态。启动实例只需很短的时间。实例启动后，其状态将更改为 `InService`。您可以看到 Auto Scaling 组启动了 1 个新实例，并且实例处于 `InService` 状态。

更改 Auto Scaling 组的大小（Amazon CLI）

更改 Auto Scaling 组的大小时，Amazon EC2 Auto Scaling 将管理启动或终止实例的过程以维护新的组大小。默认行为是不等待默认冷却时间完成，但您可以覆盖默认设置并等待冷却时间完成。有关更多信息，请参阅Amazon EC2 Auto Scaling 的缩放冷却时间 (p. 152)。

以下示例假设您创建了一个最小大小为 1，最大大小为 5 的 Auto Scaling 组。因此，该组目前正在运行的实例是 1 个。

要更改 Auto Scaling 组的大小

使用 set-desired-capacity 命令更改 Auto Scaling 组的大小，如下面的示例所示。

```
aws autoscaling set-desired-capacity --auto-scaling-group-name my-asg \
  --desired-capacity 2
```

如果您选择遵守 Auto Scaling 组的默认冷却时间，必须指定 `--honor-cooldown` 选项，如下面的示例所示。

```
aws autoscaling set-desired-capacity --auto-scaling-group-name my-asg \
  --desired-capacity 2 --honor-cooldown
```

验证 Auto Scaling 组的大小

使用 describe-auto-scaling-groups 命令确认已更改 Auto Scaling 组的大小，如下面的示例所示。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

以下是示例输出，包含关于组和已启动实例的详细信息。

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupARN": "arn",
            "ServiceLinkedRoleARN": "arn",
```

```
            "TargetGroupARNs": [],
            "SuspendedProcesses": [],
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "Tags": [],
            "EnabledMetrics": [],
            "LoadBalancerNames": [],
            "AutoScalingGroupName": "my-asg",
            "DefaultCooldown": 300,
            "MinSize": 1,
            "Instances": [
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-05b4f7d5be44822a6",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "Pending"
                },
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-0c20ac468fa3049e8",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService"
                }
            ],
            "MaxSize": 5,
            "VPCZoneIdentifier": "subnet-c87f2be0",
            "HealthCheckGracePeriod": 300,
            "TerminationPolicies": [
                "Default"
            ],
            "CreatedTime": "2019-03-18T23:30:42.611Z",
            "AvailabilityZones": [
                "us-west-2a"
            ],
            "HealthCheckType": "EC2",
            "NewInstancesProtectedFromScaleIn": false,
            "DesiredCapacity": 2
        }
    ]
}
```


注意 `DesiredCapacity` 显示了新值。Auto Scaling 组已启动另一个实例。

将 EC2 实例附上 Auto Scaling 组。

Amazon EC2 Auto Scaling 提供将一个或多个 EC2 实例附加到现有 Auto Scaling 组的选项。附加实例后，将其视为 Auto Scaling 组的一部分。

在连接实例时，请注意以下事项：

- 当您附加实例时，该组的所需容量将增加要附加的实例数。如果要附加的实例数加上所需容量超过组的最大大小，请求将失败。
- 如果您将实例附加到已附加负载均衡器目标组或经典负载均衡器的 Auto Scaling 组，则会将实例注册到该负载均衡器。

对于要附加的实例，必须满足以下条件：

- 该实例通过 Amazon EC2 处于 `running` 的状态。
- 用于启动实例的 AMI 必须仍然存在。
- 实例不是其他 Auto Scaling 组的成员。
- 实例会启动到您的 Auto Scaling 组中定义的某个可用区。
- 如果 Auto Scaling 组具有附加的负载均衡器目标组或经典负载均衡器，则实例和负载均衡器必须都位于同一 VPC 中。

附加实例（控制台）

使用以下过程将实例连接至 Auto Scaling 组。

要将实例附上现有的 Auto Scaling 组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. （可选）在导航窗格上的 Auto Scaling 下，选择 Auto Scaling Groups（Auto Scaling 组）。选择 Auto Scaling 组并验证 Auto Scaling 组的最大大小足以再添加一个实例。否则，在 Details（详细信息）选项卡上，增加最大容量。
3. 在导航窗格上的 Instances（实例）下，选择 Instances（实例），然后选择一个实例。
4. 依次选择操作、实例设置和附加到 Auto Scaling 组。
5. 在附加到 Auto Scaling 组页面上，为 Auto Scaling 组，输入组名称，然后选择附加。
6. 如果实例不符合条件，则会显示一条错误消息并提供详细信息。例如，实例可能没有位于与 Auto Scaling 组相同的可用区中。选择 Close（关闭）并使用符合条件的实例重试。

附加实例（Amazon CLI）

使用以下过程将实例连接至 Auto Scaling 组。

这些示例使用具有以下配置的 Auto Scaling 组：

- Auto Scaling 组名称 = my-asg
- 最小大小 = 2
- 最大大小 = 5
- 所需容量 = 2

要将实例附上某个 Auto Scaling 组

1. 使用 describe-auto-scaling-groups 命令描述特定的 Auto Scaling 组。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg
```

以下示例响应显示所需的容量为 2，并且组具有两个正在运行的实例。

```
{
    "AutoScalingGroups": [
```

```
        {
            "AutoScalingGroupARN": "arn",
            "ServiceLinkedRoleARN": "arn",
            "TargetGroupARNs": [],
            "SuspendedProcesses": [],
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "Tags": [],
            "EnabledMetrics": [],
            "LoadBalancerNames": [],
            "AutoScalingGroupName": "my-asg",
            "DefaultCooldown": 300,
            "MinSize": 1,
            "Instances": [
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-05b4f7d5be44822a6",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "Pending"
                },
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-0c20ac468fa3049e8",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService"
                }
            ],
            "MaxSize": 5,
            "VPCZoneIdentifier": "subnet-c87f2be0",
            "HealthCheckGracePeriod": 300,
            "TerminationPolicies": [
                "Default"
            ],
            "CreatedTime": "2019-03-18T23:30:42.611Z",
            "AvailabilityZones": [
                "us-west-2a"
            ],
            "HealthCheckType": "EC2",
            "NewInstancesProtectedFromScaleIn": false,
            "DesiredCapacity": 2
        }
    ]
}
```

2. 使用 attach-instances 命令将实例附加到 Auto Scaling 组。

```
aws autoscaling attach-instances --instance-ids i-0787762faf1c28619 --auto-scaling-
group-name my-asg
```

3. 要验证是否已附加实例，请使用如下的 describe-auto-scaling-groups 命令。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg
```

下面的示例响应显示所需容量已增加了 1 个实例（达到新容量 3），并且有一个新实例 i-0787762faf1c28619。

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupARN": "arn",
            "ServiceLinkedRoleARN": "arn",
            "TargetGroupARNs": [],
            "SuspendedProcesses": [],
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "Tags": [],
            "EnabledMetrics": [],
            "LoadBalancerNames": [],
            "AutoScalingGroupName": "my-asg",
            "DefaultCooldown": 300,
            "MinSize": 1,
            "Instances": [
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-05b4f7d5be44822a6",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "Pending"
                },
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-0c20ac468fa3049e8",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService"
                },
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
                    },
                    "InstanceId": "i-0787762faf1c28619",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService"
                }
            ],
            "MaxSize": 5,
            "VPCZoneIdentifier": "subnet-c87f2be0",
```

```
            "HealthCheckGracePeriod": 300,
            "TerminationPolicies": [
                "Default"
            ],
            "CreatedTime": "2019-03-18T23:30:42.611Z",
            "AvailabilityZones": [
                "us-west-2a"
            ],
            "HealthCheckType": "EC2",
            "NewInstancesProtectedFromScaleIn": false,
            "DesiredCapacity": 3
        }
    ]
}
```

从 Auto Scaling 组中分离 EC2 实例

您可以从 Auto Scaling 组中移除（分离）处于 `InService` 状态的实例。分离实例之后，可以独立于 Auto Scaling 组的其余部分管理这些实例。通过分离实例，您可以：

- 将实例从一个 Auto Scaling 组移出，然后将其附加到其他组。有关更多信息，请参阅将 EC2 实例附上 Auto Scaling 组。 (p. 125)。
- 使用运行应用程序的现有实例创建 Auto Scaling 组，以此来对其进行测试。随后，您可以在测试完成后从 Auto Scaling 组中分离这些实例。

在分离实例时，请注意以下事项：

- 如果要分离的实例数将 Auto Scaling 组的大小降到最小容量以下，您必须先减少该组的最小容量，然后才能分离实例。
- 分离实例时，您可以选择按分离的实例数减少 Auto Scaling 组的所需容量。如果您选择不减少容量，Amazon EC2 Auto Scaling 会启动新实例以替换分离的实例。如果减少容量，但将多个实例从同一可用区分离，则除非暂停 `AZRebalance` 流程，否则 Amazon EC2 Auto Scaling 可以重新平衡可用区。有关更多信息，请参阅暂停和恢复 Auto Scaling 组的进程 (p. 236)。
- 如果您将实例从已附加负载均衡器目标组或经典负载均衡器的 Auto Scaling 组分离，则将从该负载均衡器取消注册实例。如果为负载均衡器启用了连接耗尽（取消注册延迟），则 Amazon EC2 Auto Scaling 会等待未送达请求完成。

> **Note**
>
> 如果您要分离的实例位于 `Standby` 状态，请谨慎行事。在将实例置于 `Standby` 状态后尝试分离实例可能会导致其他实例意外终止。

分离实例（控制台）

使用以下步骤将实例与 Auto Scaling 组分离。

将实例与现有 Auto Scaling 组分离

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Instance management（实例管理）选项卡上的 Instances（实例）中，选择一个实例，然后选择 Actions（操作）、Detach（分离）。

4. 在 Detach instance（分离实例）对话框中，选中该复选框可启动替换实例，取消选中该复选框可减少所需容量。选择 Detach instance（分离实例）。

分离实例（Amazon CLI）

使用以下步骤将实例与 Auto Scaling 组分离。

这些示例使用具有以下配置的 Auto Scaling 组：

- Auto Scaling 组名称 = `my-asg`
- 最小容量 = 1
- 最大容量 = 5
- 所需容量 = 4

将实例与现有 Auto Scaling 组分离

1. 使用以下 describe-auto-scaling-instances 命令列出当前实例。

```
aws autoscaling describe-auto-scaling-instances
```

以下示例响应显示该组有 4 个正在运行的实例。

```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-05b4f7d5be44822a6",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0c20ac468fa3049e8",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0787762faf1c28619",
```

```
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0f280a4c58d319a8a",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        }
    ]
}
```

2. 使用以下 detach-instances 命令分离实例并减小所需容量。

```
aws autoscaling detach-instances --instance-ids i-05b4f7d5be44822a6 \
  --auto-scaling-group-name my-asg --should-decrement-desired-capacity
```

3. 使用以下 describe-auto-scaling-instances 命令确认已分离实例。

```
aws autoscaling describe-auto-scaling-instances
```

以下示例响应显示现在有 3 个正在运行的实例。

```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0c20ac468fa3049e8",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0787762faf1c28619",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
```

```
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-0f280a4c58d319a8a",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        }
    ]
}
```

Amazon EC2 Auto Scaling 的动态扩缩

动态扩缩会根据流量的变化扩展自动扩缩组的容量。

Amazon EC2 Auto Scaling 支持以下类型的动态扩缩策略：

- Target tracking scaling（目标跟踪扩缩）– 根据某个 Amazon CloudWatch 指标和目标值增加或减少组的当前容量。这与温控器保持家里温度的方式类似——您只需选择一个温度，剩余的工作将由温控器完成。
- Step scaling（步进分步）– 通过一系列扩缩调整（也称步进调整）来增加和减少组的当前容量，具体调整因警报严重程度而异。
- Simple scaling（简单扩缩）– 通过单次扩缩调整来增加和减少组的当前容量，每次扩缩活动之间有一个冷却时间。

如果扩缩活动所基于的指标会按比例增减自动扩缩组中的实例数量，我们建议您使用目标跟踪扩缩策略。否则，我们建议您使用步进扩展策略。

使用目标跟踪扩缩策略时，自动扩缩组会根据应用程序的实际负载按比例扩缩。这意味着，除根据负载变化满足当前容量需求外，目标跟踪策略还可以适应持续的负载变化，例如由于季节性变化而导致的负载变化。

默认情况下，新自动扩缩组在启动时没有任何扩缩策略。如果自动扩缩组不进行任何形式的动态扩展，除非您设置了计划扩展或预测性扩展，否则该组不会自行扩展。

目录

- 动态扩缩策略的工作方式 (p. 132)
- 多个动态扩缩策略 (p. 133)
- Amazon EC2 Auto Scaling 的目标跟踪扩缩策略 (p. 133)
- 有关更多信息，请参阅 Amazon EC2 Auto Scaling 的步进和简单扩展策略 (p. 141)
- 设置实例预热或扩缩冷却的原定设置值 (p. 149)
- 根据 Amazon SQS 进行扩展 (p. 155)
- 验证 Auto Scaling 组的扩缩活动 (p. 160)
- 禁用 Auto Scaling 组的扩缩策略 (p. 161)
- 删除扩展策略 (p. 163)
- Amazon Command Line Interface (Amazon CLI) 的扩展策略示例 (p. 164)

动态扩缩策略的工作方式

动态扩缩策略指示 Amazon EC2 Auto Scaling 跟踪特定 CloudWatch 指标，并定义当关联的 CloudWatch 警报处于 ALARM 状态时要采取的操作。用于调用警报状态的指标是来自自动扩缩组中所有实例的指标聚合。（例如，假设您有一个 Auto Scaling 组，其中有两个实例，一个实例的 CPU 利用率为 60%，另一个实例的 CPU 利用率为 40%。CPU 平均利用率为 50%。）策略生效后，在超过警报的阈值时，Amazon EC2 Auto Scaling 向上或向下调整组的所需容量。

如果调用动态扩缩策略，容量计算生成的数字超出了组的最小和最大大小范围，则 Amazon EC2 Auto Scaling 确保新容量永远不会超出最小和最大大小限制。容量通过以下两种方式衡量：使用您在以实例数设置所需容量时选择的相同单位；或者使用容量单位（如果应用了实例权重 (p. 67)）。

- 示例 1：Auto Scaling 组的最大容量为 3，当前容量为 2，并有增加 3 个实例的动态扩缩策略。调用此策略后，Amazon EC2 Auto Scaling 仅向组添加 1 个实例，以防止组超出其最大大小。
- 示例 2：Auto Scaling 组的最小容量为 2，当前容量为 3，并有移除 2 个实例的动态扩缩策略。调用执行此策略后，Amazon EC2 Auto Scaling 仅从组中移除 1 个实例，以防止组小于其最小大小。

当所需容量达到最大大小限制时，向外扩展停止。如果需求下降并且容量下降，则 Amazon EC2 Auto Scaling 会再次扩展。

但使用实例权重时例外。在这种情况下，Amazon EC2 Auto Scaling 可以扩展超过最大大小限制，但上限为您的最大实例权重。其目的是尽可能接近新的所需容量，但仍然遵循为该组指定的分配战略。分配策略决定要启动的实例类型。权重根据实例类型确定每个实例向所需的组容量贡献的容量单位数。

- 示例 3：Auto Scaling 组的最大容量为 12，当前容量为 10，并有增加 5 个容量单位的动态扩缩策略。向实例类型分配三个权重之一：1、4 或 6。调用策略后，Amazon EC2 Auto Scaling 根据分配策略，选择启动权重为 6 的实例类型。此横向扩展事件的结果是得到所需容量为 12、当前容量为 16 的组。

多个动态扩缩策略

在大多数情况下，目标跟踪扩缩策略就足以将您的 Auto Scaling 组配置为自动扩展和缩减。目标跟踪扩缩策略允许您选择所需结果，并让 Auto Scaling 组根据需要添加和删除实例以实现该结果。

对于高级扩缩配置，您的 Auto Scaling 组可以有多个扩缩策略。例如，您可以定义一个或多个目标跟踪扩展策略，一个或多个步进扩展策略，或者同时使用两种策略。这样可以更灵活地覆盖多种场景。

为了说明多个扩缩策略如何协同工作，请设想一个应用程序，它使用一个 Auto Scaling 组和 Amazon SQS 队列将请求发送到单个 EC2 实例。为了帮助确保应用程序性能达到最佳级别，有两个策略用于控制何时扩缩 Auto Scaling 组。一个是使用自定义指标、根据队列中的 SQS 消息数增加和移除容量的目标跟踪策略。另一个是步进扩缩策略，它使用 Amazon CloudWatch `CPUUtilization` 指标，当实例在指定时间长度内的使用率超过 90% 时增加容量。

如果同时实施多个策略，各个策略可能会同时指示 Auto Scaling 组扩展（或缩减）。例如，在 SQS 自定义指标达到峰值并超过自定义指标警报的阈值时，`CPUUtilization` 指标可能会达到峰值并超过 CloudWatch 警报的阈值。

如果发生上述情况，Amazon EC2 Auto Scaling 会选择在扩展和缩减时都提供最大容量的策略。例如，假定 `CPUUtilization` 策略启动一个实例，而 SQS 队列的策略启动两个实例。如果同时满足两个策略的扩展条件，Amazon EC2 Auto Scaling 优先选择 SQS 队列策略。这会导致 Auto Scaling 组启动两个实例。

即使策略使用不同的扩展条件，使提供最大容量的策略具有优先级的方法也适用。例如，如果一个策略终止三个实例，另一个策略将实例数量减少 25%，且在缩减时组中有八个实例，则 Amazon EC2 Auto Scaling 会优先考虑为组提供最大数量实例的策略。这会导致 Auto Scaling 组终止两个实例（8 X 25% = 2）。目的是防止 Amazon EC2 Auto Scaling 删除过多实例。

不过，在将目标跟踪扩展策略与步进扩展策略结合使用时，我们建议您务必谨慎，因为这些策略之间的冲突可能会导致意外的行为。例如，如果步进扩展策略在目标跟踪策略准备执行缩减之前启动缩减活动，则不会阻止缩减活动。在缩减活动完成后，目标跟踪策略可能会指示组重新横向扩展。

Amazon EC2 Auto Scaling 的目标跟踪扩缩策略

要创建目标跟踪扩缩策略，您需要指定一个 Amazon CloudWatch 指标和一个目标值，目标值用来表示应用程序的理想平均利用率或吞吐量水平。然后 Amazon EC2 Auto Scaling 可以横向扩展该组（增加实例）以处理高峰流量，并在利用率或吞吐量较低时横向缩减该组（减少运行的实例数量）以降低成本。

例如，假设您当前有一个在两个实例上运行的 Web 应用程序，并希望在应用程序负载变化时将自动扩缩组的 CPU 利用率保持在 50% 左右。这为您提供额外容量以处理流量高峰，而无需维护过多的空闲资源。

创建一个将目标平均 CPU 利用率设置为 50% 的目标跟踪扩缩策略即可满足此需求。然后，您的自动扩缩组会根据需要增减实例数量，以确保实际指标值等于或接近 50%。

主题

- 多个目标跟踪扩缩策略 (p. 134)
- 注意事项 (p. 134)
- 选择指标 (p. 135)
- 定义目标值 (p. 136)
- 定义实例预热时间 (p. 136)
- 创建目标跟踪扩缩策略（控制台） (p. 136)
- 创建目标跟踪扩展策略 (Amazon CLI) (p. 137)
- 使用指标数学为 Amazon EC2 Auto Scaling 创建目标跟踪扩展策略 (p. 139)

多个目标跟踪扩缩策略

为帮助优化扩缩性能，您可以将多个目标跟踪扩缩策略组合使用，但前提是其中的每个策略都各自使用不同的指标。例如，利用率和吞吐量可能会相互影响。每当其中一个指标发生变化时，通常意味着其他指标也将受到影响。因此，使用多个指标可以提供有关自动扩缩组所承担负载的额外信息，并改进有关需要向组中增加的容量的决策。

Amazon EC2 Auto Scaling 的目的是始终优先考虑可用性，因此其行为会有所不同，具体取决于目标跟踪策略是否已准备好扩展或缩减。如果任何目标跟踪策略已准备好进行扩展，它将扩展 Auto Scaling 组，但仅在所有目标跟踪策略（启用了缩减部分）准备好缩减时才执行缩减。

注意事项

使用目标跟踪扩缩策略时，需要注意以下事项：

- 请勿创建、编辑或删除与目标跟踪扩缩策略一起使用的 CloudWatch 警报。Amazon EC2 Auto Scaling 会创建并管理与您的目标跟踪扩缩策略关联的 CloudWatch 警报，并在不需要时将其删除。
- 在流量波动时，目标跟踪扩缩策略会优先考虑可用性，在流量减少时以更缓步的方式横向缩减。如果您希望自动扩缩组在工作负载完成后立即横向缩减，则可以禁用策略的横向缩减部分。这使您能够灵活地使用最适合您需求的横向缩减方法。为确保尽快进行横向缩减，我们建议不要使用简单扩缩策略以防止增加冷却时间。
- 如果指标缺少数据点，则会导致 CloudWatch 警报状态变为 `INSUFFICIENT_DATA`。发生这种情况时，在找到新的数据点之前，Amazon EC2 Auto Scaling 无法扩展您的组。
- 您可能会看到目标值与实际指标数据点之间存在差距。这是因为我们在确定需要添加或删除多少个实例时通过向上或向下取整来保守地进行操作，以免添加的实例数量不足或删除的实例数量过多。但是，对于实例较少的 Auto Scaling 组，组的使用率可能偏离目标值较远。例如，假设您将 CPU 使用率的目标值设置为 50%，然后 Auto Scaling 组超过了该目标。我们可以确定，添加 1.5 个实例会将 CPU 使用率降低到接近 50%。由于不可能添加 1.5 个实例，我们将该值向上取整，添加两个实例。这可能会将 CPU 使用率降至 50% 以下，但可确保应用程序具有充足的支持资源。同样，如果我们确定删除 1.5 个实例可使 CPU 使用率提高到 50% 以上，我们将只删除一个实例。
- 对于包含更多实例的更大 Auto Scaling 组，使用率分布在更多实例上，在这种情况下，添加或删除实例会导致目标值与实际指标数据点之间的差距缩小。
- 目标跟踪扩缩策略假设它应该在指定指标高于目标值时扩展 Auto Scaling 组。在指定指标低于目标值时，不能使用目标跟踪扩缩策略来横向扩展自动扩缩组。

选择指标

在目标跟踪扩展策略中，您可以使用预定义指标或自定义指标。

可供使用的预定义指标如下：

- `ASGAverageCPUUtilization`—Auto Scaling 组的平均 CPU 利用率。
- `ASGAverageNetworkIn` – 单个实例在所有网络接口上收到的平均字节数。
- `ASGAverageNetworkOut` – 单个实例在所有网络接口上发送的平均字节数。
- `ALBRequestCountPerTarget` – 每目标的应用程序负载均衡器请求计数。

> **Important**
>
> 对于 CPU 利用率、网络 I/O 和每目标的应用程序负载均衡器请求计数等指标，有关其他的重要信息，请分别参见《适用于 Linux 实例的 Amazon EC2 用户指南》中的列出实例的可用 CloudWatch 指标主题和《适用于应用程序负载均衡器的用户指南》中的应用程序负载均衡器的 CloudWatch 指标主题。

您可以选择其他可用的 CloudWatch 指标，也可以在 CloudWatch 中指定自定义指标来选择您自己的指标。您必须使用 Amazon CLI 或开发工具包，创建包含自定义指标的目标跟踪策略。

选择指标时请记住原则：

- 并非所有指标都适用于目标跟踪。当指定自定义指标时，这可能很重要。指标必须是有效的使用率指标，它用于描述实例的繁忙程度。指标值必须随着 Auto Scaling 组中的实例数按比例增加或减少。这样，指标数据可用于随实例数量按比例扩展或缩减。例如，如果某个 Auto Scaling 组的负载分布在各个实例上，则该 Auto Scaling 组的 CPU 使用率指标（即指标维度为 `AutoScalingGroupName` 的 Amazon EC2 指标 `CPUUtilization`）能够正常工作。
- 以下指标不适用于目标跟踪：
  - Auto Scaling 组前面的负载均衡器收到的请求数（即，Elastic Load Balancing 指标 `RequestCount`）。负载均衡器收到的请求数不会根据 Auto Scaling 组的使用率而发生变化。
  - 负载均衡器请求延迟（即，Elastic Load Balancing 指标 `Latency`）。请求延迟可能会根据使用率增加而增加，但不一定按比例变化。
  - CloudWatch Amazon SQS 队列指标 `ApproximateNumberOfMessagesVisible`。队列中的消息数可能不会随着处理队列中的消息的 Auto Scaling 组的大小按比例发生变化。但是，用于测量消息数（自动扩缩组的每个 EC2 实例的队列中）的自定义指标能够正常工作。有关更多信息，请参阅根据 Amazon SQS 进行扩展 (p. 155)。
- 要使用 `ALBRequestCountPerTarget` 指标，您必须指定 `ResourceLabel` 参数以标识与该指标关联的负载均衡器目标组。
- 指标向 CloudWatch 发出实际 0 值时（例如，`ALBRequestCountPerTarget`），自动扩缩组可在应用程序没有流量时可以横向缩减至 0。要在没有请求路由时将自动扩缩组横向缩减至 0，组的最小容量必须设置为 0。

在扩缩策略中使用 EC2 实例指标时，我们建议您以 1 分钟的粒度配置这些指标，以确保更快地响应指标值的变化。如果以 5 分钟的粒度根据实例指标扩缩，可能会导致响应速度变慢，并根据过时的指标数据进行扩缩。

要获取此级别的 Amazon EC2 指标数据，您必须专门启用详细监控功能。默认情况下，EC2 实例针对基本监控启用，也就是说，实例的指标数据以 5 分钟一次的粒度提供。有关更多信息，请参阅配置 Auto Scaling 实例的监控 (p. 253)。

定义目标值

创建目标跟踪扩缩策略时，必须指定一个目标值。目标值表示自动扩缩组的最佳平均利用率或吞吐量。为了经济高效地使用资源，目标值的设置应尽可能高，并为流量的意外增加提供合的理缓冲。当应用程序针对正常流量进行最佳横向扩展时，实际指标值应等于或略低于目标值。

当扩缩策略基于吞吐量（例如，每目标的应用程序负载均衡器请求计数、网络 I/O 或其他计数指标）时，目标值表示单个实例在一分钟内的最佳平均吞吐量。

定义实例预热时间

Important

我们建议使用默认实例预热设置，该设置统一了自动扩缩组的所有预热和冷却设置。有关更多信息，请参阅可用的预热和冷却设置 (p. 154)。

您可以指定新启动实例的预热时间（单位为秒）。在指定的预热时间过期前，实例不会计入自动扩缩组的聚合 EC2 实例指标。

当实例处于预热时间时，仅当未预热实例的指标值大于策略的目标利用率时，您的扩缩策略才会横向扩展。

如果组再次横向扩展，则仍在预热的实例将计算为下一横向扩展活动所需容量的一部分。旨在持续 (但不过度) 扩大。

当横向扩展活动正在进行时，通过扩缩策略启动的所有横向缩减活动都将被阻止，直到实例完成预热。

创建目标跟踪扩缩策略（控制台）

您可以选择在创建 Auto Scaling 组时或在创建 Auto Scaling 组之后对其配置目标跟踪扩缩策略。

要创建具有目标跟踪扩缩策略的 Auto Scaling 组

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选择 Create Auto Scaling group（创建 Auto Scaling 组）。
3. 在步骤 1、2 和 3 中，根据需要选择选项，然后继续步骤 4：配置组大小和扩缩策略。
4. 在组大小下方，通过更新最小容量和最大容量来指定扩缩范围。这两个设置允许您的 Auto Scaling 组动态扩缩。Amazon EC2 Auto Scaling 按最小容量和最大容量指定的值范围扩缩您的组。
5. 在扩缩策略下方，选择目标跟踪扩缩策略。
6. 要定义策略，请执行以下操作：
   a. 指定策略的名称。
   b. 对于 Metric type（指标类型），选择一个指标。

      如果您选择了每个目标的 Application Load Balancer 请求计数，请在目标组中选择目标组。
   c. 为指标指定 Target value（目标值）。
   d. （可选）对于 Instances need（实例需要），请根据需要更新实例预热值。
   e. （可选）选择 Disable scale in to create only a scale-out policy（禁用缩减以创建仅扩展策略）。这样，可以根据需要创建独立的其他类型的缩减策略。
7. 继续创建 Auto Scaling 组。您的扩展策略将在创建 Auto Scaling 组后创建。

为现有 Auto Scaling 组创建目标跟踪扩展策略

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。

2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 验证是否正确设置了最小和最大容量。例如，如果组已经是最大大小，则指定一个新的最大值才能进行扩展。Amazon EC2 Auto Scaling 不会超出最小或最大容量范围。要更新组，请在 Details（详细信息）选项卡上，更改最小容量和最大容量的当前设置。
4. 在 Automatic scaling（自动扩展）选项卡的 Dynamic scaling policies（动态扩展策略）中，选择 Create dynamic scaling policy（创建动态扩展策略）。
5. 要定义策略，请执行以下操作：

   a. 对于 Policy type（策略类型），保留默认的 Target tracking scaling（目标跟踪扩展）。

   b. 指定策略的名称。

   c. 对于 Metric type（指标类型），选择一个指标。您只能选择一种指标类型。要使用多个指标，请创建多个策略。

      如果您选择了每个目标的 Application Load Balancer 请求计数，请在目标组中选择目标组。

   d. 为指标指定 Target value（目标值）。

   e. （可选）对于 Instances need（实例需要），请根据需要更新实例预热值。

   f. （可选）选择 Disable scale in to create only a scale-out policy（禁用缩减以创建仅扩展策略）。这样，可以根据需要创建独立的其他类型的缩减策略。
6. 选择 Create（创建）。

创建目标跟踪扩展策略 (Amazon CLI)

按如下方式使用 Amazon CLI 为 Auto Scaling 组配置目标跟踪扩展策略。

任务

- 第 1 步：创建 Auto Scaling 组 (p. 137)
- 步骤 2：创建目标跟踪扩展策略 (p. 137)

第 1 步：创建 Auto Scaling 组

使用 create-auto-scaling-group 命令并利用启动模板 `my-template` 创建名为 `my-asg` 的自动扩缩组。如果您没有启动模板，请参阅 使用启动模板的 Amazon CLI 示例 (p. 29)。

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateName=my-template,Version='2' \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --max-size 5 --min-size 1
```

步骤 2：创建目标跟踪扩展策略

在您创建 Auto Scaling 组后，可以创建目标跟踪扩展策略，指示 Amazon EC2 Auto Scaling 在应用程序负载变化时动态地增加或减少组中正在运行的 EC2 实例数量。

示例：目标跟踪配置文件

以下是将 CPU 平均使用率保持在 40% 的示例目标跟踪配置。将此配置保存在名为 `config.json` 的文件中。

```
{
```

```
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
    {
      "PredefinedMetricType": "ASGAverageCPUUtilization"
    }
}
```

有关更多信息，请参阅 Amazon EC2 Auto Scaling API 参考中的 PredefinedMetricSpecification。

或者，您可以通过创建自定义指标规范并为 CloudWatch 的每个参数添加值来使用自定义指标进行扩缩。以下是一个示例目标跟踪配置，它将指定指标的平均利用率保持在 40%。

```
{
   "TargetValue":40.0,
   "CustomizedMetricSpecification":{
      "MetricName":"MyUtilizationMetric",
      "Namespace":"MyNamespace",
      "Dimensions":[
         {
            "Name":"MyOptionalMetricDimensionName",
            "Value":"MyOptionalMetricDimensionValue"
         }
      ],
      "Statistic":"Average",
      "Unit":"Percent"
   }
}
```

有关更多信息，请参阅 Amazon EC2 Auto Scaling API 参考中的 CustomizedMetricSpecification。

示例：cpu40-target-tracking-scaling-policy

使用 put-scaling-policy 命令及之前创建的 `config.json` 文件创建一个名为 `cpu40-target-tracking-scaling-policy` 的扩展策略，用于将 Auto Scaling 组的 CPU 平均使用率保持在 40%：

```
aws autoscaling put-scaling-policy --policy-name cpu40-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
```

如果成功，此命令将返回包含代表您创建的两个 CloudWatch 警报的 ARN 和名称。

```
{
    "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:228f02c2-c665-4bfd-
aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/cpu40-target-tracking-scaling-
policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-my-asg-
AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
            "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-my-asg-
AlarmLow-61a39305-ed0c-47af-bd9e-471a352ee1a2",
            "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
        }
    ]
}
```

使用指标数学为 Amazon EC2 Auto Scaling 创建目标跟踪扩展策略

借助指标数学，您可以查询多个 CloudWatch 指标并使用数学表达式来创建基于这些指标的新时间序列。您可以在 CloudWatch 控制台中直观显示生成的时间序列，并将其添加到控制面板中。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用指标数学。

以下考虑因素适用于指标数学表达式：

- 您可以查询任何可用的 CloudWatch 指标。每个指标都是指标名称、命名空间和零个或多个维度的唯一组合。
- 您可以使用任意算术运算符 (+ - * / ^)、统计函数（例如 AVG 或 SUM）或 CloudWatch 支持的其他函数。
- 您可以在数学表达式的公式中同时使用指标和其他数学表达式的结果。
- 指标规范中使用的任何表达式最终都必须返回一个单个时间序列。
- 您可以使用 CloudWatch 控制台或 CloudWatch GetMetricData API 验证指标数学表达式是否有效。

> **Note**
>
> 只有在使用 Amazon CLI 或开发工具包时您才能使用指标数学创建目标跟踪扩展策略。此功能在控制台和 Amazon CloudFormation 中尚不可用。

示例：每个实例的 Amazon SQS 队列积压

要计算每个实例的 Amazon SQS caling 队列积压，请获取可用于从队列中检索的消息的大致数量，然后将该数字除以自动扩缩组的运行容量，该数字即为处于 `InService` 状态的实例数量。有关更多信息，请参阅根据 Amazon SQS 进行扩展 (p. 155)。

表达式的逻辑如下：

```
sum of (number of messages in the queue)/(number of InService instances)
```

然后，您的 CloudWatch 指标信息如下所示。

ID	CloudWatch 指标	统计数据	周期
m1	ApproximateNumberOfMessagesVisible	总计	1 minute
m2	GroupInServiceInstances	平均值	1 minute

您的指标数学 ID 和表达式如下所示。

ID	表达式
e1	`(m1)/(m2)`

使用该指标数学来创建目标跟踪扩展策略 (Amazon CLI)

1. 将指标数学表达式作为自定义指标规范的一部分存储在名为 `config.json` 的 JSON 文件中。

   使用下面的示例帮助您快速开始。将`#######`更换为适用于您的应用程序的值。

```
{
    "CustomizedMetricSpecification": {
        "Metrics": [
```

```
            {
                "Label": "Get the queue size (the number of messages waiting to be
 processed)",
                "Id": "m1",
                "MetricStat": {
                    "Metric": {
                        "MetricName": "ApproximateNumberOfMessagesVisible",
                        "Namespace": "AWS/SQS",
                        "Dimensions": [
                            {
                                "Name": "QueueName",
                                "Value": "my-queue"
                            }
                        ]
                    },
                    "Stat": "Sum"
                },
                "ReturnData": false
            },
            {
                "Label": "Get the group size (the number of InService instances)",
                "Id": "m2",
                "MetricStat": {
                    "Metric": {
                        "MetricName": "GroupInServiceInstances",
                        "Namespace": "AWS/AutoScaling",
                        "Dimensions": [
                            {
                                "Name": "AutoScalingGroupName",
                                "Value": "my-asg"
                            }
                        ]
                    },
                    "Stat": "Average"
                },
                "ReturnData": false
            },
            {
                "Label": "Calculate the backlog per instance",
                "Id": "e1",
                "Expression": "m1 / m2",
                "ReturnData": true
            }
        ]
    },
    "TargetValue": 100
}
```

有关更多信息，请参阅《Amazon EC2 Auto Scaling API 参考》中的 TargetTrackingConfiguration。

Note

以下是一些可以帮助您查找 CloudWatch 指标的指标名称、命名空间、维度和统计数据的其他资源：

- 有关 Amazon 服务可用指标的更多信息，请参阅 Amazon CloudWatch 用户指南中的发布 CloudWatch 指标的 Amazon 服务。
- 要获取带有 Amazon CLI 的 CloudWatch 指标的确切指标名称、命名空间和维度（如果适用），请参阅列出指标.

2. 要创建此策略，请运行 put-scaling-policy 命令并将此 JSON 文件作为输入，如下例所示。

```
aws autoscaling put-scaling-policy --policy-name sqs-backlog-target-tracking-scaling-
policy \
```

```
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
```

如果成功，则此命令将返回策略的 Amazon 资源名称（ARN）和代表您创建的两个 CloudWatch 告警的 ARN。

```
{
    "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:228f02c2-
c665-4bfd-aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/sqs-backlog-target-
tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-my-
asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
            "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-my-
asg-AlarmLow-61a39305-ed0c-47af-bd9e-471a352ee1a2",
            "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
        }
    ]
}
```

Note

如果此命令引发错误，请确保您已在本地将 Amazon CLI 更新到最新版本。

有关更多信息，请参阅 Amazon EC2 Auto Scaling 的步进和简单扩展策略

通过步进扩展和简单扩展，您可以为触发扩展过程的 CloudWatch 警报选择扩展指标和阈值。您还可以定义在指定数目的评估期内违反阈值时应如何扩展 Auto Scaling 组。

我们强烈建议您使用目标跟踪扩展策略，根据类似于平均 CPU 利用率等指标或 Application Load Balancer 的 `RequestCountPerTarget` 指标来扩展。使用目标跟踪，可以通过在容量增加时减少以及在容量减少时增加的指标，按比例扩展或缩减实例数。这有助于确保 Amazon EC2 Auto Scaling 密切遵循应用程序的需求曲线。有关更多信息，请参阅目标跟踪扩缩策略 (p. 133)。您仍然可以选择使用步进扩展作为附加策略来实现更高级的配置。例如，您可以在利用率达到特定级别时配置更积极的响应。

目录

- 步进扩展策略和简单扩展策略之间的区别 (p. 142)
- 步进扩展的分步调整 (p. 142)
- 扩展调整类型 (p. 143)
- 实例预热 (p. 144)
- 为指标的高、低阈值创建 CloudWatch 警报（控制台） (p. 144)
- 创建步骤扩展策略（控制台） (p. 146)
- 创建扩展策略和 CloudWatch 警报 (Amazon CLI) (p. 147)
    - 第 1 步：创建 Auto Scaling 组 (p. 147)
    - 第 2 步：创建扩展策略 (p. 147)
        - 分步扩展策略 (p. 147)
        - 简单扩展策略 (p. 148)

- 步骤 3：为指标的高、低阈值创建 CloudWatch 警报 (p. 149)

步进扩展策略和简单扩展策略之间的区别

步进扩展策略和简单扩展策略是可供您使用的两个动态扩展选项。两者都要求您为扩展策略创建 CloudWatch 警报。两者都要求您为警报指定高阈值和低阈值。这均需要您定义添加还是删除实例及实例的数量，或者将组设置为确切的大小。

策略类型之间的主要区别在于通过步进扩展策略获得的步进调整。当应用步进调整并且增加或减少了 Auto Scaling 组的当前容量时，调整会根据警报违规的大小而有所不同。

在大多数情况下，步进扩展策略是比简单扩展策略更好的选择，即使您仅有单个扩展调整也是如此。

简单扩缩的主要问题是，在扩缩活动开始后，该策略必须等待扩缩活动或运行状况检查替换完成并且冷却时间 (p. 152)到期，然后才会响应其他警报。冷却时间有助于防止在先前活动产生明显影响前启动其他扩展活动。

与之对比，步进扩展策略可以继续响应其他警报，甚至在进行扩展活动或运行状况检查替换时也是如此。因此，Amazon EC2 Auto Scaling 会在收到警报消息时评估任何违例警报。

Amazon EC2 Auto Scaling 起初只支持简单扩展策略。如果您在引入目标跟踪和分步策略前创建了自己的扩展策略，您的策略将被视为简单扩展策略。

步进扩展的分步调整

在创建分步扩展策略时，您可以指定一个或多个步进调整，这些步进调整会根据警报违规的大小自动扩展实例数量。每个分步调整指定以下内容：

- 指标值的下限
- 指标值的上限
- 要扩展的数量（基于扩展调整类型）

CloudWatch 根据与您的 CloudWatch 警报关联的指标的统计信息聚合指标数据点。超过警报时，将调用相应的扩缩策略。Amazon EC2 Auto Scaling 将指定的聚合类型应用于 CloudWatch 的最新指标数据点（而不是原始指标数据）。它将此聚合指标值与步进调整定义的上限和下限进行比较，以确定执行哪个步进调整。

您可以指定相对于违例阈值的上限和下限。例如，假设您有一个既具有当前容量又具有所需容量 10 的 Auto Scaling 组。您有一个违规阈值为 50% 的 CloudWatch 警报以及扩展和缩减策略。对于每个策略，您有一组调整类型为 `PercentChangeInCapacity`（或控制台中的 % 的组）的步进调整：

示例：扩展策略的步进调整

下限	上限	调整
0	10	0
10	20	10
20	null	30

示例：缩放策略的步进调整

下限	上限	调整
-10	0	0

下限	上限	调整
-20	-10	-10
null	-20	-30

这将创建以下扩展配置。

```
Metric value

-infinity          30%    40%          60%     70%             infinity
-------------------------------------------------------------------------
          -30%      | -10% | Unchanged  | +10%  |       +30%
-------------------------------------------------------------------------
```

以下几点总结了扩展配置相对于组的所需容量和当前容量的行为：

- 在聚合指标值大于 40 并小于 60 时，将保留所需容量和当前容量。
- 如果指标值达到 60，根据扩展策略的第二个分步调整 (增加 10 个实例的 10%)，组的所需容量将增加 1 个实例，总共达到 11 个实例。在新实例运行并且其指定预热时间过期后，当前组容量将增加到 11 个实例。如果指标值即使在这一容量增加之后也上升到 70，则组的所需容量将再增加 3 个实例，达到 14 个实例。这基于扩展策略的第三个步进调整（添加 11 个实例的 30%，即 3.3 个实例，向下舍入到 3 个实例）。
- 如果指标值降到 40，根据缩减策略的第二个分步调整 (删除 14 个实例的 10%，向下取整为 1 个实例)，组的所需容量将减小 1 个实例，达到 13 个实例。如果指标值即使在这一容量减少之后也下降至 30，则组的所需容量将再减少 3 个实例，达到 10 个实例。这基于缩减策略的第三个步进调整（移除 13 个实例的 30%，即 3.9 个实例，向下舍入为 3 个实例）。

为扩展策略指定步进调整时，请注意以下事项：

- 如果您使用 Amazon Web Services Management Console，则以绝对值的形式指定上限和下限。如果您使用 Amazon CLI 或开发工具包，则以违例阈值相对值的形式指定上限和下限。
- 分步调整范围不能重叠或有间隙。
- 只有一个分步调整可以有空下限 (负无穷)。如果一个分步调整有负下限，则必须有一个分步调整有空下限。
- 只有一个分步调整可以有空上限 (正无穷)。如果一个步进调整有正上限，则必须有一个步进调整有空上限。
- 同一步进调整中的上限和下限不能为空。
- 如果指标值高于违例阈值，则含下限而不含上限。如果指标值低于违例阈值，则不含下限而含上限。

扩展调整类型

您可以根据您选择的扩展调整类型来定义执行最佳扩展操作的扩展策略。您可以将调整类型指定为 Auto Scaling 组当前容量的百分比或以容量单位指定。通常，一个容量单位意味着一个实例，除非您使用的是实例权重功能。

对于步进扩展和简单扩展，Amazon EC2 Auto Scaling 支持以下调整类型：

- `ChangeInCapacity` — 将当前组容量增加或减少指定的值。正值将增加容量，负调整值将减小容量。例如：如果组的当前容量为 3，调整值为 5，那么当执行此策略时，我们会向容量添加 5 个容量单位，共计 8 个容量单位。
- `ExactCapacity` — 将组的当前容量更改为指定值。为该调整类型指定一个正值。例如：如果当前组容量为 3，调整值为 5，则当执行此策略时，我们会将容量更改为 5 个容量单位。

- `PercentChangeInCapacity` — 将当前组容量增加或减少指定的百分比。正值将增加容量，负值将减少容量。例如：如果当前容量为 10，调整值为 10%，那么当执行此策略时，我们会向容量添加 1 个容量单位，总共 11 个容量单位。

  Note

  如果得出的值不是整数，将按如下所示进行取整：
  - 大于 1 的值向下取整。例如，`12.7` 取整为 `12`。
  - 0 和 1 之间的值舍入到 1。例如，`.67` 取整为 `1`。
  - 0 和 -1 之间的值舍入到 -1。例如，`-.58` 取整为 `-1`。
  - 小于 -1 的值向上取整。例如，`-6.67` 取整为 `-6`。

通过 `PercentChangeInCapacity`，您还可以使用 `MinAdjustmentMagnitude` 参数指定要扩展的最小实例。例如，假定您创建一个增加 25% 的策略，并且您指定最小增量为 2 个实例。如果您有一个 Auto Scaling 组包含 4 个实例，而要执行该扩展策略，则 4 的 25% 就是 1 个实例。但是，因为您指定了最小增量 2，则将添加 2 个实例。

使用实例权重 (p. 67)时，将 `MinAdjustmentMagnitude` 参数设置为非零值的效果会发生变化。该值以容量单位为单位。如需设置所要扩展的最小实例数，请将此参数设置为至少与最大实例权重一样大的值。

如果您使用实例权重，请记住，Auto Scaling 组的当前容量可以根据需要超过所需容量。如果要递减的绝对数或百分比表示递减的数量小于当前容量和所需容量之间的差值，则不会执行任何扩展操作。在您查看超过警报阈值时扩缩策略的结果时，必须考虑这些行为。例如，假设所需容量为 30，当前容量为 32。超过警报时，如果扩缩策略将所需容量减少 1，则不会执行扩缩操作。

实例预热

Important

我们建议使用 `DefaultInstanceWarmup` 设置，它统一了 Auto Scaling 组的所有预热和冷却设置。有关更多信息，请参阅可用的预热和冷却设置 (p. 154)。

对于扩缩步骤，您可以指定新启动实例的预热时间（单位为秒）。在指定的预热时间过期前，实例不会计入自动扩缩组的聚合 EC2 实例指标。

当实例处于预热时间时，仅当未预热实例的指标值大于策略的告警阈值上限时，您的扩缩策略才会横向扩展。

如果组再次横向扩展，则仍在预热的实例将计算为下一横向扩展活动所需容量的一部分。因此，落入同一分步调整中的多个警报违例只会导致一个扩展活动。旨在持续 (但不过度) 扩大。

当横向扩展活动正在进行时，通过扩缩策略启动的所有横向缩减活动都将被阻止，直到实例完成预热。

为指标的高、低阈值创建 CloudWatch 警报（控制台）

您可以使用 CloudWatch 控制台创建两个警报，一个用于横向扩展（指标的高阈值），另一个用于横向缩减（指标的低阈值）。例如，当超过警报的阈值时，由于流量增加，警报将进入 ALARM (警报) 状态。此状态更改会调用与警报关联的扩缩策略。然后，该策略指示 Amazon EC2 Auto Scaling 如何响应“超过警报”事件，例如添加或删除指定数量的实例。

示例：要为指标的高阈值创建 CloudWatch 警报

1. 通过以下网址打开 CloudWatch 控制台：https://console.aws.amazon.com/cloudwatch/。
2. 如果需要，可以更改区域。从导航栏中，选择您的自动扩缩组所在的区域。
3. 在导航窗格中，选择 Alarms, All alarms（警报，所有警报），然后选择 Create alarm（创建警报）。

4. 选择选择指标。
5. 在 All metrics（所有指标）选项卡上，选择 EC2、By Auto Scaling Group（按 Auto Scaling 组），然后在搜索字段中输入 Auto Scaling 组的名称。然后，选择 `CPUUtilization` 并选择 Select metric（选择指标）。将显示 Specify metric and conditions（指定指标和条件）页面，其中显示一个图表以及有关指标的其他信息。
6. 在 Period（周期）下，选择警报的评估周期，例如 1 分钟。评估警报时，每个周期都聚合到一个数据点。

   Note

   周期越短，创建的警报越敏感。
7. 在 Conditions（条件）下，执行以下操作：
   - 对于 Threshold type（阈值类型），选择 Static（静态）。
   - 对于每当 **`CPUUtilization`** 为，请指定您是希望指标值大于、大于等于、小于还是小于等于阈值以触发警报。然后，在 than（大于/小于）下，输入您希望超过警报的阈值。

     Important

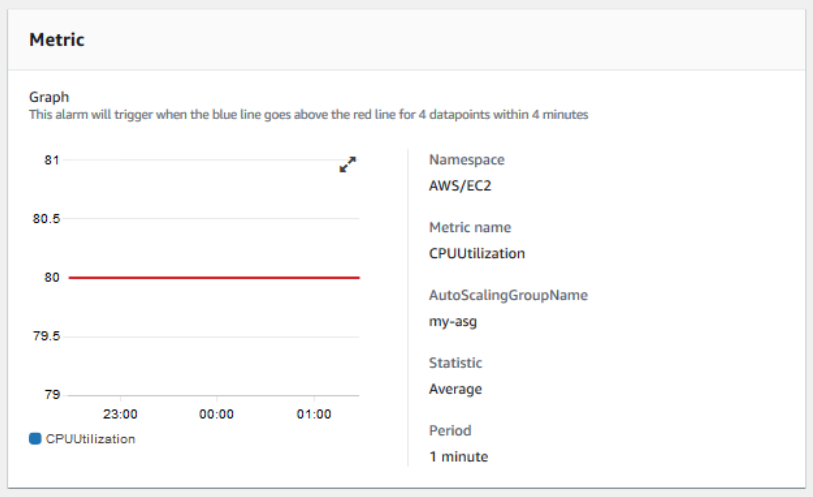
     要使警报与横向扩展策略（指标的高阈值）一起使用，请确保不要选择小于或小于等于阈值。要使警报与横向缩减策略（警报的低阈值）一起使用，请确保不要选择大于或大于等于阈值。
8. 在其他配置下，执行以下操作：
   - 对于 Datapoints to alarm（触发警报的数据点数），输入指标值必须满足阈值条件才会触发警报的数据点（评估时间段）数。例如，2 个连续的 5 分钟时间段需要花 10 分钟才会调用警报状态。
   - 对于 Missing data treatment（缺失数据处理），选择 Treat missing data as bad (breaching threshold)（将丢失的数据视为不良数据（违反阈值））。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的配置 CloudWatch 警报处理缺少数据的方式。
9. 选择 Next（下一步）。

   Configure actions（配置操作）页面会显示。
10. 在 Notification（通知）下面，选择一个在警报处于 `ALARM`、`OK` 或 `INSUFFICIENT_DATA` 状态时通知的 Amazon SNS 主题。

    要使告警为相同告警状态或不同告警状态发送多个通知，请选择 Add notification（添加通知）。

    要让警报不发送通知，请选择删除。
11. 您可以保留 Configure actions（配置操作）页面的其他部分为空。将其他部分留空会创建警报，而不会将其与扩展策略相关联。然后，您可以从 Amazon EC2 Auto Scaling 控制台将警报与扩展策略关联。
12. 选择 Next（下一步）。
13. 输入警报的名称（例如，`Step-Scaling-AlarmHigh-AddCapacity`）和可选的描述，然后选择 Next（下一步）。
14. 选择Create alarm（创建警报）。

示例：如果 CPU 超出阈值 80% 达 4 分钟，则超出 CloudWatch 警报



创建步骤扩展策略（控制台）

创建自动扩缩组后，您可以选择在该组上配置分步扩缩策略。

以下过程说明如何使用 Amazon EC2 Auto Scaling 控制台创建两个步进扩展策略：将组容量增加 30% 的扩展策略，以及将组容量减小到两个实例的收缩策略。

在配置您的扩展策略时，您可以同时创建警报。或者，您也可以使用从 CloudWatch 控制台创建的警报，如上一部分中所述。

为向外扩展创建步进扩展策略

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 验证是否正确设置了最小和最大大小限制。例如，如果组已经是最大大小，则需要指定一个新的最大值才能进行扩展。Amazon EC2 Auto Scaling 不会超出最小或最大容量范围。要更新组，请在 Details（详细信息）选项卡上，更改最小容量和最大容量的当前设置。
4. 在 Automatic scaling（自动扩展）选项卡的 Dynamic scaling policies（动态扩展策略）中，选择 Create dynamic scaling policy（创建动态扩展策略）。
5. 要为向外扩展（增加容量）定义策略，请执行以下操作：

   a. 对于 Policy type（策略类型），选择 Step scaling（步进扩展）。

   b. 指定策略的名称。

   c. 对于 CloudWatch alarm (CloudWatch 警报)，请选择您的警报。如果尚未创建警报，请选择 Create a CloudWatch alarm（创建 CloudWatch 警报），然后在 为指标的高、低阈值创建 CloudWatch 警报（控制台）(p. 144) 中完成步骤 4 到步骤 14 以创建监控 CPU 利用率的警报。将警报阈值设置为大于或等于 80%。

   d. 指定在使用 Take the action (执行操作) 来完成操作时，此策略对当前组大小进行的更改。您可以添加特定数量的实例或现有组大小的百分比，也可将组设置为准确的大小。

      例如，选择 Add，在下一个字段中输入 30，然后选择 `percent of group`。默认情况下，此步骤调整的下限为警报阈值，上限为正 (+) 无穷。

   e. 要添加另一个步骤，请选择 Add step (添加步进)，然后定义要缩放的量以及步进相对于警报阈值的下限和上限。

f. 要设置可扩展的最少实例数，请更新 Add capacity units in increments of at least (添加容量单位的增量至少为) 1 capacity units (容量单位) 中的数量字段。

g. （可选）对于 Instances need（实例需要），请根据需要更新实例预热值。

6. 选择 Create（创建）。

为缩减创建步进扩展策略

1. 选择创建动态扩展策略以在为向外扩展创建策略之后继续之前的操作。
2. 要为缩减（减少容量）定义策略，请执行以下操作：

   a. 对于 Policy type（策略类型），选择 Step scaling（步进扩展）。

   b. 指定策略的名称。

   c. 对于 CloudWatch alarm (CloudWatch 警报)，请选择您的警报。如果尚未创建警报，请选择 Create a CloudWatch alarm（创建 CloudWatch 警报），然后在 为指标的高、低阈值创建 CloudWatch 警报（控制台）(p. 144) 中完成步骤 4 到步骤 14 以创建监控 CPU 利用率的警报。将警报阈值设置为小于或等于 40%。

   d. 指定在使用 Take the action (执行操作) 来完成操作时，此策略对当前组大小进行的更改。您可以删除特定数量的实例或现有组大小的百分比，也可将组设置为准确的大小。

      例如，选择 `Remove`，在下一个字段中输入 2，然后选择 `capacity units`。默认情况下，此步骤调整的上限为警报阈值，下限为负 (-) 无穷。

   e. 要添加另一个步骤，请选择 Add step (添加步进)，然后定义要缩放的量以及步进相对于警报阈值的下限和上限。

3. 选择 Create（创建）。

创建扩展策略和 CloudWatch 警报 (Amazon CLI)

按如下方式使用 Amazon CLI，为您的 Auto Scaling 组配置步进扩展策略或简单扩展策略。

任务

- 第 1 步：创建 Auto Scaling 组 (p. 147)
- 第 2 步：创建扩展策略 (p. 147)
- 步骤 3：为指标的高、低阈值创建 CloudWatch 警报 (p. 149)

第 1 步：创建 Auto Scaling 组

使用以下 create-auto-scaling-group 命令并利用启动模板 `my-template` 创建名为 `my-asg` 的自动扩缩组。如果您没有启动模板，请参阅 使用启动模板的 Amazon CLI 示例 (p. 29)。

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateName=my-template,Version='2' \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --max-size 5 --min-size 1
```

第 2 步：创建扩展策略

您可以创建分步或简单扩展策略，指示 Auto Scaling 在应用程序负载发生变化时该怎么做。

分步扩展策略

示例：my-step-scale-out-policy

使用以下 put-scaling-policy 命令创建名为 my-step-scale-out-policy 的分步扩缩策略，其调整类型为 PercentChangeInCapacity，基于以下分步调整增加组容量（假设 CloudWatch 警报阈值为 60%）：

- 当指标值大于或等于 60% 但小于 75% 时，将实例计数增加 10%
- 当指标值大于或等于 75% 但小于 85% 时，将实例计数增加 20%
- 当指标值大于或等于 85% 时，将实例计数增加 30%

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg  \
  --policy-name my-step-scale-out-policy \
  --policy-type StepScaling \
  --adjustment-type PercentChangeInCapacity \
  --metric-aggregation-type Average \
  --step-adjustments
 MetricIntervalLowerBound=0.0,MetricIntervalUpperBound=15.0,ScalingAdjustment=10 \

 MetricIntervalLowerBound=15.0,MetricIntervalUpperBound=25.0,ScalingAdjustment=20 \
                    MetricIntervalLowerBound=25.0,ScalingAdjustment=30 \
  --min-adjustment-magnitude 1
```

记下策略的 Amazon Resource Name (ARN)。您需要使用它来为策略创建 CloudWatch 警报。

```
{
    "PolicyARN": "arn:aws:autoscaling:region:123456789012:scalingPolicy:4ee9e543-86b5-4121-
b53b-aa4c23b5bbcc:autoScalingGroupName/my-asg:policyName/my-step-scale-in-policy
}
```

示例：my-step-scale-in-policy

使用以下 put-scaling-policy 命令创建名为 my-step-scale-in-policy 的分步扩缩策略，调整类型为 ChangeInCapacity，并且在关联的 CloudWatch 警报超过指标的低阈值时将组容量减少 2 个实例。

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg  \
  --policy-name my-step-scale-in-policy \
  --policy-type StepScaling \
  --adjustment-type ChangeInCapacity \
  --step-adjustments MetricIntervalUpperBound=0.0,ScalingAdjustment=-2
```

记下策略的 Amazon Resource Name (ARN)。您需要使用它来为策略创建 CloudWatch 警报。

```
{
    "PolicyARN": "arn:aws:autoscaling:region:123456789012:scalingPolicy:ac542982-
cbeb-4294-891c-a5a941dfa787:autoScalingGroupName/my-asg:policyName/my-step-scale-out-policy
}
```

简单扩展策略

或者，您可以使用以下 CLI 命令而不是前文中的 CLI 命令创建简单扩展策略。请记住，使用简单扩展策略会取代冷却时间。

示例：my-simple-scale-out-policy

使用以下 put-scaling-policy 命令创建名为 my-simple-scale-out-policy 的简单扩缩策略，调整类型为 PercentChangeInCapacity，并且在关联的 CloudWatch 警报超过指标的高阈值时将组容量增加 30%。

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-out-policy \
```

```
  --auto-scaling-group-name my-asg --scaling-adjustment 30 \
  --adjustment-type PercentChangeInCapacity
```

记下策略的 Amazon Resource Name (ARN)。您需要使用它来为策略创建 CloudWatch 警报。

示例：my-simple-scale-in-policy

使用以下 put-scaling-policy 命令创建名为 `my-simple-scale-in-policy` 的简单扩缩策略，调整类型为 `ChangeInCapacity`，并且在关联的 CloudWatch 警报超过指标的低阈值时将组容量减少一个实例。

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-in-policy \
  --auto-scaling-group-name my-asg --scaling-adjustment -1 \
  --adjustment-type ChangeInCapacity --cooldown 180
```

记下策略的 Amazon Resource Name (ARN)。您需要使用它来为策略创建 CloudWatch 警报。

步骤 3：为指标的高、低阈值创建 CloudWatch 警报

在步骤 2 中，您已创建扩缩策略，这些策略可以指示 Amazon EC2 Auto Scaling 在指标值增加或降低时如何横向扩展和横向缩减。在此步骤中，您可以创建两个警报，方法为标识要监控的指标、为警报定义指标的高、低阈值和其他详细信息，然后将警报与扩缩策略关联。

示例：AddCapacity

使用以下 CloudWatch put-metric-alarm 命令创建警报，基于至少两个连续评估期（两分钟）内 60% 的平均 CPU 阈值增加 Auto Scaling 组的大小。要使用您自己的自定义指标，请在 `--metric-name` 中指定其名称，并在 `--namespace` 指定其命名空间。

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmHigh-AddCapacity \
  --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average \
  --period 120 --evaluation-periods 2 --threshold 60 \
  --comparison-operator GreaterThanOrEqualToThreshold \
  --dimensions "Name=AutoScalingGroupName,Value=my-asg" \
  --alarm-actions PolicyARN
```

示例：RemoveCapacity

使用以下 CloudWatch put-metric-alarm 命令创建警报，基于至少两个连续评估期（两分钟）内 40% 的平均 CPU 阈值减小 Auto Scaling 组的大小。要使用您自己的自定义指标，请在 `--metric-name` 中指定其名称，并在 `--namespace` 指定其命名空间。

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmLow-RemoveCapacity \
  --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average \
  --period 120 --evaluation-periods 2 --threshold 40 \
  --comparison-operator LessThanOrEqualToThreshold \
  --dimensions "Name=AutoScalingGroupName,Value=my-asg" \
  --alarm-actions PolicyARN
```

设置实例预热或扩缩冷却的原定设置值

原定设置实例预热和原定设置冷却是可以用来优化扩缩性能的 Auto Scaling 组属性。

目录

- 为 Auto Scaling 组设置原定设置实例预热 (p. 150)
- Amazon EC2 Auto Scaling 的缩放冷却时间 (p. 152)

- 可用的预热和冷却设置 (p. 154)

为 Auto Scaling 组设置原定设置实例预热

您可以使用原定设置实例预热功能来优化用于动态扩缩的 Amazon CloudWatch 指标。CloudWatch 会跨 Auto Scaling 实例采集和聚合使用情况数据，例如 CPU 和网络 I/O。您可以使用这些指标来创建扩缩策略，以随着所选指标值的增减调整 Auto Scaling 组中的实例数量。

当未启用原定设置实例预热时，每个实例将在实例到达 `InService` 状态时立即向聚合指标提供使用情况数据。但如果启用了原定设置实例预热，则实例将在完成预热后才会提供使用情况数据。从而防止动态扩缩受到尚未处理应用程序流量，并且可能暂时存在计算资源使用率较高的单个实例的指标影响。

预设情况下不会配置和启用原定设置实例预热。为了优化会持续扩展的扩缩策略（例如目标跟踪扩缩策略和步进扩缩策略）的性能，我们强烈建议您启用原定设置实例预热功能，即使其值设置为 0 秒。如果未启用原定设置实例预热，则可能会在实例启动时执行原定设置冷却，动态扩缩的横向缩减活动将被阻止，直到原定设置冷却时间结束。

默认实例预热将多个功能的预热参数统一为组级别的单个参数。通过启用默认实例预热功能，您不必再为以下功能指定预热参数值，除非您想覆盖默认：

- 实例刷新 (p. 95)
- 目标跟踪扩展策略 (p. 136)
- 分步扩展策略 (p. 144)

除了影响扩展性能外，默认实例预热还会影响在实例刷新期间替换实例所需的总时间。

目录

- 选择原定设置的实例预热时间值 (p. 150)
- 了解原定设置实例预热如何影响扩缩 (p. 150)
- 为组启用原定设置实例预热 (p. 151)
- 验证组的原定设置实例预热设置 (p. 152)

选择原定设置的实例预热时间值

选择预热时间值时，应注意优化平衡采集合法流量的使用数据与尽量减少与启动时的临时使用峰值相关的数据采集。

完成启动后，实例将附加到 Auto Scaling 组并注册到 Elastic Load Balancing 负载均衡器（如果使用），然后才会进入 `InService` 状态。在实例进入 `InService` 状态之后，资源消耗仍然可能会经历暂时的高峰，然后才会逐渐稳定下来。例如，与无需下载大型资产的轻量级 Web 服务器相比，对于必须下载并缓存大型资产的应用程序服务器，其资源消耗将需要更长的时间才能稳定。原定设置实例预热提供了稳定资源消耗所需的时间延迟。

> **Important**
>
> 对于需要在启动时运行配置任务或脚本的使用案例，则还应考虑使用生命周期钩子。在横向扩展事件持续期间，生命周期钩子可以将实例投入使用的时间延迟到实例完成初始化之后。如果引导启动脚本需要一段时间才能完成，则生命周期钩子将特别有用。因此，如果您添加了生命周期钩子，则可以降低原定设置实例预热的值。如果您将原定设置实例预热设为 0 秒，则不会在生命周期钩子所需时间之外，强加额外的等待时间。有关使用生命周期钩子的更多信息，请参阅 Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

了解原定设置实例预热如何影响扩缩

启用原定设置实例预热后，Auto Scaling 组中每个新启动的实例都有一个预热时间。

当实例处于预热时间时，仅当未预热实例的指标值大于策略的告警阈值上限（与目标跟踪扩缩策略的目标利用率相同）时，您的扩缩策略才会横向扩展。

如果实例仍在预热并且组再次横向扩展，则这些实例将计算为下一横向扩展活动所需容量的一部分。因此，落入同一分步调整中的多个警报违例只会导致一个扩展活动。旨在持续 (但不过度) 扩大。

其目的是在需求下降时谨慎地横向缩减，以保护应用程序的可用性。这会阻止通过扩缩策略启动的横向缩减活动，直到实例完成预热。

为组启用原定设置实例预热

您可以在创建 Auto Scaling 组时启用原定设置实例预热。也可以为现有的组启用此功能。

Console

为新的组启用原定设置实例预热（控制台）

创建 Auto Scaling 组时，在 Configure advanced options（配置高级选项）页面的 Additional settings（其他设置）下，选择 Enable default instance warmup（启用原定设置实例预热）选项。选择应用程序所需的预热时间。

Amazon CLI

为新的组启用原定设置实例预热（Amazon CLI）

要为 Auto Scaling 组启用原定设置实例预热，请添加 `--default-instance-warmup` 选项并指定一个介于 0 到 3600 之间的值（以秒为单位）。启用此功能后，将值设为 `-1` 将会关闭此设置。

以下 create-auto-scaling-group 命令会创建一个名称为 *my-asg* 的 Auto Scaling 组，同时会启用原定设置实例预热功能，并将其值设置为 *120* 秒。

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --default-
instance-warmup 120 ...
```

Tip

如果此命令引发错误，请确保您已在本地将 Amazon CLI 更新到最新版本。

Console

为现有的组启用原定设置实例预热（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 在屏幕顶部的导航栏中，选择您在其中创建了自动扩缩组的 Amazon Web Services 区域。
3. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
4. 在 Details（详细信息）选项卡上，选择 Advanced configurations（高级配置）、Edit（编辑）。
5. 对于 Default instance warmup（原定设置实例预热），选择应用程序所需的预热时间。
6. 选择 Update（更新）。

Amazon CLI

为现有的组启用原定设置实例预热（Amazon CLI）

以下示例使用 update-auto-scaling-group 命令为名为的 *my-asg* 的现有 Auto Scaling 组启用原定设置实例预热，并将其值设置为 *120* 秒。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --default-
instance-warmup 120
```

Tip

如果此命令引发错误，请确保您已在本地将 Amazon CLI 更新到最新版本。

验证组的原定设置实例预热设置

验证 Auto Scaling 组的原定设置实例预热设置（Amazon CLI）

使用 describe-auto-scaling-groups 命令。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

以下为响应示例。

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            ...
            "DefaultInstanceWarmup": 120
        }
    ]
}
```

Amazon EC2 Auto Scaling 的缩放冷却时间

在您的 Auto Scaling 组启动或终止实例后，它首先等待冷却时间结束，然后才能启动简单扩缩策略启动的任何扩缩活动。冷却时间的用途是防止 Auto Scaling 组在先前的活动产生明显的作用之前启动或终止其他实例。

Important

作为最佳实践，我们建议您不要使用简单扩缩策略和扩缩冷却。
在大多数情况下，目标跟踪扩缩策略或步进扩缩策略更能提升扩缩性能。对于随着扩缩指标值的减小或增加而按比例更改 Auto Scaling 组大小的扩缩策略，我们建议采用目标跟踪 (p. 133)而不是简单扩缩或分步扩缩。

例如，假设一个有关 CPU 利用率的简单扩缩策略建议启动两个实例。Amazon EC2 Auto Scaling 会启动两个实例，然后暂停扩缩活动，直到冷却时间结束。冷却时间结束后，简单扩缩策略启动的任何扩缩活动都可以恢复。如果 CPU 利用率再次超过告警阈值上限，则 Auto Scaling 组将再次横向扩展，而冷却时间也会再次生效。不过，如果两个实例足以将指标值降为正常水平，该组会保持其当前大小。

目录

- 注意事项 (p. 153)
- 生命周期钩子会导致额外的延迟 (p. 153)
- 更改原定设置冷却时间 (p. 153)
- 为特定的简单扩缩策略设置冷却时间 (p. 153)

注意事项

在使用简单扩缩策略和扩缩冷却时，应注意以下事项：

- 目标跟踪扩缩策略和步进扩缩策略可以立即启动横向扩展活动，而无需等待冷却时间结束。相反，每当您的 Auto Scaling 组启动实例时，单个实例都有一个预热时间。有关更多信息，请参阅为 Auto Scaling 组设置原定设置实例预热 (p. 150)。
- 当计划的操作在计划的时间开始时，它还可能会立即启动扩缩活动，而无需等待冷却时间结束。
- 如果实例运行不正常，Amazon EC2 Auto Scaling 不会等到冷却时间结束才替换运行不正常的实例。
- 当多个实例启动或终止时，冷却时间（无论是原定设置冷却还是特定于扩缩策略的冷却）会从最后一个实例完成启动或终止时开始生效。
- 当您手动扩展 Auto Scaling 组时，预设情况下不会等待冷却时间结束。但您可以覆盖此行为，并在使用 Amazon CLI 或者 SDK 手动扩展时遵守此原定设置冷却设置。
- 预设情况下，Elastic Load Balancing 会等待 300 秒，以便完成注销（Connection Draining）过程。如果组位于 Elastic Load Balance 负载均衡器后面，它将等待终止的实例完成注销，然后再开始计算冷却时间。

生命周期钩子会导致额外的延迟

如果触发了生命周期钩子 (p. 191)，则冷却时间将在您完成生命周期操作后或超时时间结束后开始计算。例如，假设某个 Auto Scaling 组具有一个有关实例启动的生命周期钩子。当应用程序需求增加时，该组会启动实例以增加容量。由于存在生命周期钩子，所以实例将置于等待状态，并且由简单扩缩策略触发的扩缩活动将被暂停。当实例进入 `InService` 状态时，冷却时间开始。冷却时间结束后，将恢复简单扩缩策略活动。

启用 Elastic Load Balancing 时，对于横向缩减，冷却时间将从正在终止的实例完成 Connection Draining（注销延迟）之时开始计算。它不会等待钩子的持续时间结束。这意味着，横向缩减事件的结果反映在组的容量中后，因简单扩缩策略触发的任何扩缩活动都可以立即恢复。否则，等待所有三个活动（Connection Draining、生命周期钩子和冷却时间）完成会显著增加 Auto Scaling 组暂停扩缩所需的时间量。

更改原定设置冷却时间

首次在 Amazon EC2 Auto Scaling 控制台中创建 Auto Scaling 组时，您将无法设定原定设置冷却。预设情况下，此冷却时间设置为 300 秒（5 分钟）。如果需要，您可以在创建组后更新此设置。

更改原定设置冷却时间（控制台）

创建 Auto Scaling 组后，在 Details（详细信息）选项卡上，依次选择Advanced configurations（高级配置）、Edit（编辑）。对于 Default cooldown（原定设置冷却），请根据实例启动时间或其他应用程序需求选择所需的时长。

更改原定设置冷却时间（Amazon CLI）

使用以下命令更改新的或现有 Auto Scaling 组的原定设置冷却。如果未定义原定设置冷却，则使用 300 秒的原定设置。

- create-auto-scaling-group
- update-auto-scaling-group

要确认原定设置冷却的值，请使用 describe-auto-scaling-groups 命令。

为特定的简单扩缩策略设置冷却时间

预设情况下，所有简单扩缩策略都使用为 Auto Scaling 组定义的原定设置冷却时间。要设置特定简单扩缩策略的冷却时间，请在创建或更新策略时使用可选的冷却参数。在为策略指定冷却时间时，它将覆盖原定设置冷却。

对于扩缩策略特定的冷却时间，一个常见的使用场景是横向缩减策略。由于此策略是要终止实例，Amazon EC2 Auto Scaling 需要较短的时间来确定是否终止其他实例。终止实例应该是比启动实例快得多的操作。因此，300 秒的默认冷却时间太长。在这种情况下，为横向缩减策略设置一个具有较低值的扩缩策略特定冷却时间，可让该组更快横向缩减，从而帮助降低成本。

要在控制台中创建或更新简单扩缩策略，请在创建该组之后选择 Automatic scaling（弹性伸缩）选项卡。要使用 Amazon CLI 创建或更新简单扩缩策略，请使用 put-scaling-policy 命令。有关更多信息，请参阅步进和简单扩展策略 (p. 141)。

可用的预热和冷却设置

为了帮助优化扩缩性能，请为您的 Auto Scaling 组选择适当的预热和冷却设置。

我们建议使用 `DefaultInstanceWarmup` 设置，它统一了所有预热和冷却设置。如果需要，您也可以使用 `HealthCheckGracePeriod` 设置。

其他可用设置不宜与 `DefaultInstanceWarmup` 设置同时使用。例如，已经定义了 `DefaultInstanceWarmup` 时将不需要 `EstimatedInstanceWarmup` 和 `InstanceWarmup`，而 `DefaultCooldown` 和 `Cooldown` 仅在使用简单扩缩策略时才需要。（作为最佳实践，我们建议您使用目标跟踪扩缩策略和步进扩缩策略，而不是简单扩缩策略。）但我们将继续支持所有这些设置，以便您可以根据自己的使用案例选择适当的设置。

`DefaultInstanceWarmup` (推荐)

API 操作：CreateAutoScalingGroup、UpdateAutoScalingGroup

在新实例进入 `InService` 状态后，认为它已经完成初始化和资源消耗变得稳定之前所需的时间（以秒为单位）。

在实例刷新期间，Amazon EC2 Auto Scaling 在替换一个实例之后等待预热期，然后再继续替换下一个实例。Amazon EC2 Auto Scaling 也会等待预热期，然后将新实例的指标与用于扩展的 Amazon CloudWatch 指标中的现有实例聚合在一起，从而获得更加可靠的使用数据。有关更多信息，请参阅为 Auto Scaling 组设置原定设置实例预热 (p. 150)。

如果您使用生命周期挂钩准备新的实例以供使用，则可以降低预热期的值。有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

> **Important**
>
> 要在组级别管理各预热设置，我们建议您设置默认实例预热，即使其值设置为 0 秒。要移除先前设置的值，请包含该属性，但应将该值指定为 `-1`。但我们强烈建议您通过将值指定为 `0` 或其他标称值来保持启用默认实例预热。

默认值：无

`EstimatedInstanceWarmup`

定义了 *`DefaultInstanceWarmup`* 时不需要。

API 操作：PutScalingPolicy

新启动的实例可影响 CloudWatch 指标的估计时间（以秒为单位）。此预热时间适用于因特定目标跟踪扩缩策略或步进扩缩策略启动的实例。在此处指定预热时间时，它将覆盖原定设置实例预热。有关更多信息，请参阅 目标跟踪扩缩策略 (p. 133) 和 步进和简单扩展策略 (p. 141)：

原定设置：如果未定义，则此参数的值默认为为该组定义的原定设置实例预热的值。如果原定设置实例预热为空，则将回退到原定设置冷却时间的值。

InstanceWarmup

定义了 *DefaultInstanceWarmup* 时不需要。

API 操作：StartInstanceRefresh

一段时间（以秒为单位），在此期间，在新实例进入 InService 状态后，实例刷新在继续替换下一个实例之前等待。指定启动实例刷新时的预热时间将覆盖原定设置实例预热，但仅适用于当前的实例刷新。有关更多信息，请参阅基于实例刷新替换 Auto Scaling 实例 (p. 95)。

原定设置：如果未定义，则此参数的值默认为为该组定义的原定设置实例预热的值。如果原定设置实例预热为空，则将回退为运行状况检查宽限期的值。

DefaultCooldown

仅在使用简单扩缩策略时才需要。

API 操作：CreateAutoScalingGroup、UpdateAutoScalingGroup

从因简单扩缩策略触发的一个扩缩活动结束到另一个扩缩活动开始之间的时长（秒）。有关更多信息，请参阅Amazon EC2 Auto Scaling 的缩放冷却时间 (p. 152)。

原定设置：300 秒。

Cooldown

仅在使用简单扩缩策略时才需要。

API 操作：PutScalingPolicy

适用于特定简单扩缩策略的冷却时间（以秒为单位）。（不再建议使用简单扩缩策略。作为最佳实践，我们建议改为目标跟踪扩缩策略和步进扩缩策略。）在此处指定冷却时间时，它将覆盖原定设置冷却。有关更多信息，请参阅Amazon EC2 Auto Scaling 的缩放冷却时间 (p. 152)。

默认值：无

HealthCheckGracePeriod

API 操作：CreateAutoScalingGroup、UpdateAutoScalingGroup

Amazon EC2 Auto Scaling 在检查已投入使用的 EC2 实例的运行状况，并且会因 Elastic Load Balancing 或自定义运行状况检查失败而将其标记为运行不正常之前需要等待的时间（以秒为单位）。如果您的实例在进入 InService 状态后未立即通过这些运行状况检查，则这将非常有用。有关更多信息，请参阅设置自动扩缩组的运行状况检查宽限期 (p. 246)。

如果您为实例启动使用了生命周期钩子，则可以将运行状况检查宽限期的值设置为 0。如果您的 Auto Scaling 组位于负载均衡器后面，则应考虑向组添加生命周期钩子，以确保实例在生命周期钩子结束时注册到负载均衡器之前，已准备好提供流量。有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

原定设置：在 Amazon Web Services Management Console 中创建的 Auto Scaling 组为 300 秒。使用 Amazon CLI 或 SDK 创建的 Auto Scaling 组为 0 秒。

根据 Amazon SQS 进行扩展

> Important
>
> 以下信息和步骤向您展示了如何在将每个实例的 Amazon SQS 队列积压作为自定义指标发布到 CloudWatch 前使用 ApproximateNumberOfMessages 队列属性计算它。但是，您现在可以使

用指标数学来节省发布自己的指标所花费的成本和精力。有关更多信息，请参阅使用指标数学为 Amazon EC2 Auto Scaling 创建目标跟踪扩展策略 (p. 139)。

本节介绍如何扩展 Auto Scaling 组以响应 Amazon Simple Queue Service (Amazon SQS) 队列中系统负载的变化。要了解有关如何使用 Amazon SQS 的更多信息，请参阅 Amazon 简单队列服务开发人员指南。

在一些情况下，您可能会考虑根据 Amazon SQS 队列中的活动扩展。例如，假设您有 Web 应用程序，让用户上传图像并联机使用。在此场景中，每个图像需要先调整大小并编码，然后才能发布。该应用程序在 Auto Scaling 组中的 EC2 实例上运行，配置为处理您的典型上传速率。不正常的实例将终止并进行替换，以始终保持当前的实例等级。该应用程序将图像的原始位图数据放在 SQS 队列中以进行处理。它处理这些图像，然后将处理的图像发布到某个位置以供用户查看。如果上传的图像数不随时间波动，则适用于此场景的架构可以良好运作。但是，如果上传数量随着时间波动，您可以考虑使用动态扩展来缩放 Auto Scaling 组的容量。

目录

- 将目标跟踪与合适的指标结合使用 (p. 156)
- 限制和先决条件 (p. 157)
- 配置基于 Amazon SQS 的缩放 (p. 157)
- Amazon SQS 和实例缩减保护 (p. 159)

将目标跟踪与合适的指标结合使用

如果您使用基于自定义 Amazon SQS 队列指标的目标跟踪扩展策略，则动态扩展可以更有效地适应应用程序的需求曲线。有关为目标跟踪选择指标的更多信息，请参阅 选择指标 (p. 135)。

为目标跟踪使用 CloudWatch Amazon SQS 指标（例如 `ApproximateNumberOfMessagesVisible`）的问题是，队列中的消息数，可能不会随着处理队列中消息的 Auto Scaling 组的大小按比例发生变化。这是因为 SQS 队列中的消息数不仅仅定义所需的实例数。Auto Scaling 组中的实例数可能由多种因素决定，包括处理消息所需的时间以及可接受的延迟长度（队列延迟）。

该解决方案使用要维护的每个实例的积压 指标以及每个实例的可接受积压 的目标值。您可以按以下所示计算这些数字：

- 每个实例的积压：要计算您每个实例的积压，首先通过 `ApproximateNumberOfMessages` 度列属性确定 SQS 队列的长度（可从队列中检索的消息数）。将该数字除以队列的运行容量（对于 Auto Scaling 组，这是处于 `InService` 状态的实例数量），以获得每个实例的积压。
- 每个实例的可接受积压：要计算您的目标值，请先确定您的应用程序可以接受的延迟。然后，将可接受的延迟值除以 EC2 实例处理一条消息所用的平均时间。

例如，假设您当前有一个包含 10 个实例的自动扩缩组，并且队列中的可见消息数量（`ApproximateNumberOfMessages`）是 1500。如果每条消息的平均处理时间为 0.1 秒，最长可接受延迟为 10 秒，则每个实例的可接受积压为 10/0.1，即 100 条消息。这意味着您的目标跟踪策略的目标值为 100。当每个实例的积压达到目标值时，将发生横向扩展事件。由于每个实例已经积压了 150 条消息（10 个实例 1500 条消息），该组会横向扩展并增加了 5 个实例以维持与目标值的比例。

以下过程演示如何发布自定义指标和创建目标跟踪扩展策略，以根据这些计算值来配置您的 Auto Scaling 组进行扩展。

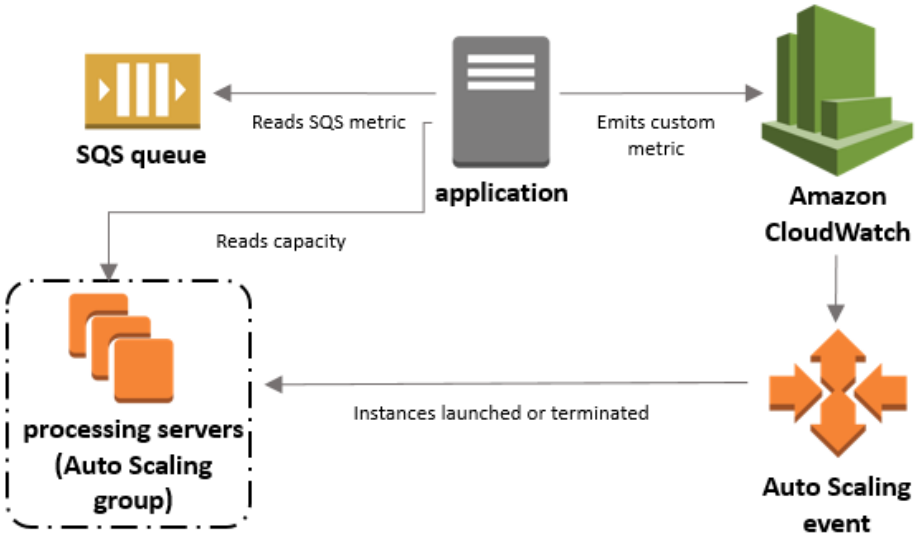
> **Important**
>
> 请记住，为了降低成本，请改用指标数学。有关更多信息，请参阅使用指标数学为 Amazon EC2 Auto Scaling 创建目标跟踪扩展策略 (p. 139)。

此配置有三个主要部分：

- 一个 Auto Scaling 组，用于管理处理来自 SQS 队列的消息的 EC2 实例。

- 一个发送到 Amazon CloudWatch 的自定义指标，用于衡量 Auto Scaling 组中每个 EC2 实例的队列的消息数。
- 一个目标跟踪策略，配置您的 Auto Scaling 组根据自定义指标和设定的目标值进行扩展。CloudWatch 警报调用该扩展策略。

下图演示了此配置的架构。



限制和先决条件

要使用此配置，您需要注意以下限制：

- 您必须使用 Amazon CLI 或 开发工具包将自定义指标发布到 CloudWatch。然后，您可以使用 Amazon Web Services Management Console 监控指标。
- Amazon EC2 Auto Scaling 控制台不支持使用自定义指标的目标跟踪扩缩策略。您必须使用 Amazon CLI 或者使用开发工具包为您的扩展策略指定自定义指标。

以下各节指示您为需要执行的任务使用 Amazon CLI。例如，要获取反映队列当前使用情况的指标数据，请使用 SQS get-queue-attributes 命令。确保您已经安装并配置 CLI。

开始之前，您必须有一个要使用的 Amazon SQS 队列。以下部分假设您已经有一个队列（标准队列或 FIFO 队列）、一个 Auto Scaling 组以及 EC2 实例（运行使用队列的应用程序）。有关 Amazon SQS 权限的更多信息，请参阅 Amazon 简单队列服务开发人员指南。

配置基于 Amazon SQS 的缩放

任务

- 步骤 1：创建 CloudWatch 自定义指标 (p. 157)
- 步骤 2：创建目标跟踪扩展策略 (p. 158)
- 步骤 3：测试扩展策略 (p. 159)

步骤 1：创建 CloudWatch 自定义指标

自定义指标是使用您选择的指标名称和命名空间定义的。自定义指标的命名空间不能以 AWS/ 开头。有关发布自定义指标的更多信息，请参阅 Amazon CloudWatch 用户指南中的发布自定义指标主题。

按照此过程，通过首先从您的 Amazon 账户读取信息来创建自定义指标。然后，计算前面的章节中建议的每个实例的积压指标。最后，按照 1 分钟的粒度将该数字发布到 CloudWatch。只要可能，我们强烈建议您按 1 分钟粒度的指标进行扩展，以确保更快地响应系统负载变化。

要创建 CloudWatch 自定义指标

1. 使用 SQS get-queue-attributes 命令获取在队列中等待的消息数 (`ApproximateNumberOfMessages`)。

```
aws sqs get-queue-attributes --queue-url https://sqs.region.amazonaws.com/123456789/
MyQueue \
  --attribute-names ApproximateNumberOfMessages
```

2. 使用 describe-auto-scaling-groups 命令获取组的运行容量，这是处于 `InService` 生命周期状态的实例数。此命令返回 Auto Scaling 组的实例及其生命周期状态。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg
```

3. 将该队列中可供检索的大致消息数量除以该组的运行容量，计算得出每个实例的积压。
4. 按照 1 分钟的粒度将结果发布为 CloudWatch 自定义指标。

   以下是示例 CLI put-metric-data 命令：

```
aws cloudwatch put-metric-data --metric-name MyBacklogPerInstance --
namespace MyNamespace \
  --unit None --value 20 --
dimensions MyOptionalMetricDimensionName=MyOptionalMetricDimensionValue
```

应用程序发出所需的指标之后，数据发送到 CloudWatch。该指标会显示在 CloudWatch 控制台中。您可以登录到 Amazon Web Services Management Console 并导航到 CloudWatch 页面来访问它。然后，通过导航到指标页面或者使用搜索框搜索指标来查看指标。有关查看指标的更多信息，请参阅 Amazon CloudWatch 用户指南 中的查看可用指标。

步骤 2：创建目标跟踪扩展策略

发布自定义指标后，使用自定义指标规范创建目标跟踪扩展策略

创建目标跟踪扩展策略

1. 使用以下命令可以在您的主目录的 `config.json` 文件中为扩展策略指定目标值。

   对于 `TargetValue`，计算每个实例的可接受积压指标并在此处输入。要计算此数值，请考虑正常延迟值并将其除以处理一条消息所需的平均时间。

```
$ cat ~/config.json
{
   "TargetValue":100,
   "CustomizedMetricSpecification":{
      "MetricName":"MyBacklogPerInstance",
      "Namespace":"MyNamespace",
      "Dimensions":[
         {
            "Name":"MyOptionalMetricDimensionName",
            "Value":"MyOptionalMetricDimensionValue"
         }
      ],
      "Statistic":"Average",
      "Unit":"None"
```

```
    }
}
```

2. 使用 put-scaling-policy 命令以及在之前步骤中创建的 config.json 文件来创建扩展策略：

```
aws autoscaling put-scaling-policy --policy-name sqs100-target-tracking-scaling-policy
 \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://~/config.json
```

这会创建两个警报：一个用于增加实例数量，另一个用于减少实例数量。它还将返回注册到 CloudWatch 的策略的 Amazon 资源名称（ARN），CloudWatch 使用该名称在超过指标阈值的任何时候调用扩缩。

步骤 3：测试扩展策略

在您完成设置后，验证您的扩展策略是否正常工作。您可以通过以下方式测试：增加 SQS 队列中的消息数，然后验证 Auto Scaling 组是否已启动更多 EC2 实例。您还可以通过以下方式测试：减少 SQS 队列中的消息数，然后验证 Auto Scaling 组是否终止了一个 EC2 实例。

测试扩展函数

1. 按照教程：将消息发送到 Amazon SQS 队列中的步骤将消息添加到队列。请确保您增加了队列中的消息数量，使得每个实例的积压指标超过目标值。

   您的更改调用警报可能需要几分钟时间。
2. 使用 describe-auto-scaling-groups 命令验证该组是否已启动了实例：

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

测试缩减函数

1. 按照教程：将消息发送到 Amazon SQS 队列中的步骤从队列中删除消息。请确保您减少了队列中的消息数量，使得每个实例的积压指标低于目标值。

   您的更改调用警报可能需要几分钟时间。
2. 使用 describe-auto-scaling-groups 命令验证该组是否已终止了实例：

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Amazon SQS 和实例缩减保护

实例被终止时还未处理的消息将返回 SQS 队列中，可由仍在运行的实例进行处理。对于执行长时间运行任务的应用程序，您可以选择使用实例扩展保护来控制 Auto Scaling 组扩展时终止哪些队列工作程序。

以下伪代码显示了保护长时间运行、队列驱动的工件进程免受缩放终止的一种方法。

```
while (true)
{
  SetInstanceProtection(False);
  Work = GetNextWorkUnit();
  SetInstanceProtection(True);
  ProcessWorkUnit(Work);
  SetInstanceProtection(False);
```

```
}
```

有关更多信息，请参阅实例横向缩减保护 (p. 230)。

验证 Auto Scaling 组的扩缩活动

在 Amazon EC2控制台的“Amazon EC2 Auto Scaling”部分中，适用于 Auto Scaling 组的活动记录可让您查看当前正在进行的扩缩活动的当前状态。扩展活动完成后，您可以看到它是否成功。在创建 Auto Scaling 组或向现有组添加扩展条件时，此方法特别有用。

当您向 Auto Scaling 组添加目标跟踪、步骤或简单扩展策略时，Amazon EC2 Auto Scaling 会立即开始对照该指标评估策略。在指标超过阈值达到指定数量的评估期时，指标警报将变为 ALARM（警报）状态。这意味着扩展策略可能会在创建后立即引发扩展活动。在 Amazon EC2 Auto Scaling 响应扩展策略而调整所需容量之后，您可以验证账户中的扩展活动。如果您希望从 Amazon EC2 Auto Scaling 接收告知扩展活动的电子邮件通知，请按照 在 Auto Scaling 组扩展时获取 Amazon SNS 通知 (p. 258) 中的说明操作。

> **Tip**
>
> 在以下过程中，您需要查看 Auto Scaling 组的 Activity history（活动历史记录）和 Instances（实例）部分。这两个部分都会已经显示已命名的列。要显示隐藏的列或更改显示的行数，请选择每个部分右上角的齿轮图标以打开首选项模式，根据需要更新设置，然后选择 Confirm（确认）。

要查看 Auto Scaling 组的扩缩活动（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Activity（活动）选项卡的 Activity history（活动历史记录）下，Status（状态）列显示您的 Auto Scaling 组是否已成功启动或终止实例，或者扩展活动是否仍在进行中。
4. （可选）如果有很多扩缩活动，您可以选择活动历史记录顶部边缘的 > 图标，来查看下一页的扩缩活动。
5. 在实例管理选项卡的实例下，生命周期列显示实例的状态。在实例开启并且任何生命周期钩子结束后，其生命周期状态将更改为 `InService`。Health status（运行状态）列显示对您的实例进行 EC2 实例运行状况检查的结果。

要查看 Auto Scaling 组的扩缩活动 (Amazon CLI)

使用以下 describe-scaling-activities 命令。

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

下面是示例输出。

扩展活动按开始时间排序。首先描述仍在进行的活动。

```
{
  "Activities": [
    {
      "ActivityId": "5e3a1f47-2309-415c-bfd8-35aa06300799",
      "AutoScalingGroupName": "my-asg",
      "Description": "Terminating EC2 instance: i-06c4794c2499af1df",
      "Cause": "At 2020-02-11T18:34:10Z a monitor alarm TargetTracking-my-asg-AlarmLow-
b9376cab-18a7-4385-920c-dfa3f7783f82 in state ALARM triggered policy my-target-tracking-
policy changing the desired capacity from 3 to 2.  At 2020-02-11T18:34:31Z an instance
```

```
 was taken out of service in response to a difference between desired and actual capacity,
 shrinking the capacity from 3 to 2.  At 2020-02-11T18:34:31Z instance i-06c4794c2499af1df
 was selected for termination.",
      "StartTime": "2020-02-11T18:34:31.268Z",
      "EndTime": "2020-02-11T18:34:53Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-west-2a
\"...}",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    },
...
  ]
}
```

有关输出中字段的描述，请参阅 Amazon EC2 Auto Scaling API 参考中的活动。

有关检索已删除组的扩展活动的帮助，以及有关可能会遇到的错误类型以及如何处理这些错误的信息，请参阅 Amazon EC2 Auto Scaling 问题排查 (p. 341)。

禁用 Auto Scaling 组的扩缩策略

本主题介绍如何临时禁用扩展策略，使其不会发起对 Auto Scaling 组包含的实例数进行的更改。禁用扩展策略时，配置详细信息将保留，以便您可以快速重新启用策略。相比在不需要时暂时删除策略并在以后重新创建，这种方法更容易。

在禁用扩展策略时，Auto Scaling 组不会为违反的指标警报进行扩展或缩减。但是，任何仍在进行的扩展活动都不会停止。

请注意，已禁用的扩展策略仍会计入您可以添加到 Auto Scaling 组的扩展策略数量配额中。

要禁用扩缩策略（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Automatic scaling（自动扩展）选项卡的 Dynamic scaling policies（动态扩展策略）下，选中所需扩展策略右上角的复选框。
4. 滚动到 Dynamic scaling policies（动态扩展策略）部分的顶部，然后选择 Actions（操作）、Disable（禁用）。

当您准备好重新启用扩展策略时，请重复这些步骤，然后选择 Actions（操作）、Enable（启用）。重新启用扩展策略后，如果当前有任何警报处于 ALARM 状态，您的 Auto Scaling 组可能会立即启动扩展操作。

禁用扩展策略 (Amazon CLI)

使用带有 `--no-enabled` 选项的 put-scaling-group 命令，如下所示。采用与在创建策略时相同的指定方式，在命令中指定所有选项。

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \
   --policy-name my-scaling-policy --policy-type TargetTrackingScaling \
   --estimated-instance-warmup 360 \
   --target-tracking-configuration '{ "TargetValue": 70, "PredefinedMetricSpecification":
 { "PredefinedMetricType": "ASGAverageCPUUtilization" } }' \
```

```
   --no-enabled
```

重新启用扩展策略 (Amazon CLI)

使用带有 `--enabled` 选项的 put-scaling-group 命令，如下所示。采用与在创建策略时相同的指定方式，在命令中指定所有选项。

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \
   --policy-name my-scaling-policy --policy-type TargetTrackingScaling \
   --estimated-instance-warmup 360 \
   --target-tracking-configuration '{ "TargetValue": 70, "PredefinedMetricSpecification":
 { "PredefinedMetricType": "ASGAverageCPUUtilization" } }' \
   --enabled
```

描述扩展策略 (Amazon CLI)

使用 describe-policies 命令验证扩展策略的启用状态。

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg \
   --policy-names my-scaling-policy
```

下面是示例输出。

```
{
    "ScalingPolicies": [
        {
            "AutoScalingGroupName": "my-asg",
            "PolicyName": "my-scaling-policy",
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:1d52783a-b03b-4710-
bb0e-549fd64378cc:autoScalingGroupName/my-asg:policyName/my-scaling-policy",
            "PolicyType": "TargetTrackingScaling",
            "StepAdjustments": [],
            "Alarms": [
                {
                    "AlarmName": "TargetTracking-my-asg-AlarmHigh-9ca53fdd-7cf5-4223-938a-
ae1199204502",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-9ca53fdd-7cf5-4223-938a-
ae1199204502"
                },
                {
                    "AlarmName": "TargetTracking-my-asg-AlarmLow-7010c83d-d55a-4a7a-
abe0-1cf8b9de6d6c",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-7010c83d-d55a-4a7a-
abe0-1cf8b9de6d6c"
                }
            ],
            "TargetTrackingConfiguration": {
                "PredefinedMetricSpecification": {
                    "PredefinedMetricType": "ASGAverageCPUUtilization"
                },
                "TargetValue": 70.0,
                "DisableScaleIn": false
            },
            "Enabled": true
        }
    ]
}
```

删除扩展策略

当您不再需要某个扩展策略时，可将其删除。根据扩展策略的类型，您可能还需要删除 CloudWatch 警报。删除目标跟踪扩展策略还将删除任何关联的 CloudWatch 警报。删除步进扩展策略或简单扩展策略会删除底层警报操作，但不删除 CloudWatch 警报，即使它不再具有关联的操作。

删除扩展策略（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Automatic scaling（自动扩展）选项卡的 Dynamic scaling policies（动态扩展策略）下，选中所需扩展策略右上角的复选框。
4. 滚动到 Dynamic scaling policies（动态扩展策略）部分的顶部，然后选择 Actions（操作）、Delete（删除）。
5. 当系统提示进行确认时，选择 Yes, Delete（是，删除）。
6. （可选）如果您删除步进扩展策略或简单扩展策略，请执行以下操作来删除与策略关联的 CloudWatch 警报。您可以跳过这些子步骤来保留警报以供将来使用。

   a. 通过以下网址打开 CloudWatch 控制台：https://console.aws.amazon.com/cloudwatch/。
   b. 在导航窗格上，选择 Alarms（警报）。
   c. 选择警报（例如 `Step-Scaling-AlarmHigh-AddCapacity`），然后选择 Action（操作）、Delete（删除）。
   d. 当系统提示进行确认时，选择 Delete（删除）。

要获取 Auto Scaling 组 (Amazon CLI) 的扩展策略

在您删除扩展策略之前，请使用以下 describe-policies 命令查看为 Auto Scaling 组创建了哪些扩展策略。可以在删除策略和 CloudWatch 警报时使用输出。

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg
```

可以使用 `--query` 参数按扩展策略类型筛选结果。此 `query` 语法仅在 Linux 或 macOS 上有效。在 Windows 上，请将单引号更改为双引号。

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg
  --query 'ScalingPolicies[?PolicyType==`TargetTrackingScaling`]'
```

下面是示例输出。

```
[
    {
        "AutoScalingGroupName": "my-asg",
        "PolicyName": "cpu40-target-tracking-scaling-policy",
        "PolicyARN": "PolicyARN",
        "PolicyType": "TargetTrackingScaling",
        "StepAdjustments": [],
        "Alarms": [
            {
                "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-my-
asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
                "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
```

```
            },
            {
                "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-my-
asg-AlarmLow-61a39305-ed0c-47af-bd9e-471a352ee1a2",
                "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
            }
        ],
        "TargetTrackingConfiguration": {
            "PredefinedMetricSpecification": {
                "PredefinedMetricType": "ASGAverageCPUUtilization"
            },
            "TargetValue": 40.0,
            "DisableScaleIn": false
        },
        "Enabled": true
    }
]
```

删除扩展策略 (Amazon CLI)

使用以下 delete-policy 命令。

```
aws autoscaling delete-policy --auto-scaling-group-name my-asg \
  --policy-name cpu40-target-tracking-scaling-policy
```

要删除您的 CloudWatch 控警报 (Amazon CLI)

对于步进扩展策略和简单扩展策略，请使用 delete-alarms 命令以删除与策略关联的 CloudWatch 警报。您可以跳过此步骤来保留该警报以备将来使用。您可以一次删除一个或多个警报。例如，使用以下命令可删除 Step-Scaling-AlarmHigh-AddCapacity 和 Step-Scaling-AlarmLow-RemoveCapacity 警报：

```
aws cloudwatch delete-alarms --alarm-name Step-Scaling-AlarmHigh-AddCapacity Step-Scaling-
AlarmLow-RemoveCapacity
```

Amazon Command Line Interface (Amazon CLI) 的扩展策略示例

您可以通过 Amazon Web Services Management Console、Amazon CLI 或软件开发工具包创建 Amazon EC2 Auto Scaling 扩展策略。

以下示例显示如何使用 Amazon CLI put-scaling-policy 命令为 Amazon EC2 Auto Scaling 创建扩缩策略。有关从 Amazon CLI 创建扩展策略的介绍性练习，请参阅 目标跟踪扩缩策略 (p. 133) 和 步进和简单扩展策略 (p. 141)。

示例 1：应用具有预定义指标规范的目标跟踪扩展策略

```
aws autoscaling put-scaling-policy --policy-name cpu40-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ASGAverageCPUUtilization"
  }
}
```

示例 2：应用具有自定义指标规范的目标跟踪扩展策略

```
aws autoscaling put-scaling-policy --policy-name sqs100-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
{
  "TargetValue": 100.0,
  "CustomizedMetricSpecification": {
    "MetricName": "MyBacklogPerInstance",
    "Namespace": "MyNamespace",
    "Dimensions": [{
      "Name": "MyOptionalMetricDimensionName",
      "Value": "MyOptionalMetricDimensionValue"
    }],
    "Statistic": "Average",
    "Unit": "None"
  }
}
```

示例 3：为仅向外扩展应用目标跟踪扩展策略

```
aws autoscaling put-scaling-policy --policy-name alb1000-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
{
  "TargetValue": 1000.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ALBRequestCountPerTarget",
    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-
group/943f017f100becff"
  },
  "DisableScaleIn": true
}
```

示例 4：为向外扩展应用步进扩展策略

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg  \
  --policy-name my-step-scale-out-policy \
  --policy-type StepScaling \
  --adjustment-type PercentChangeInCapacity \
  --metric-aggregation-type Average \
  --step-adjustments
 MetricIntervalLowerBound=10.0,MetricIntervalUpperBound=20.0,ScalingAdjustment=10 \

 MetricIntervalLowerBound=20.0,MetricIntervalUpperBound=30.0,ScalingAdjustment=20 \
                     MetricIntervalLowerBound=30.0,ScalingAdjustment=30 \
  --min-adjustment-magnitude 1
```

记下策略的 Amazon Resource Name (ARN)。创建 CloudWatch 警报时，您需要 ARN。

示例 5：为缩减应用步进扩展策略

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg  \
  --policy-name my-step-scale-in-policy \
  --policy-type StepScaling \
  --adjustment-type ChangeInCapacity \
  --step-adjustments MetricIntervalUpperBound=0.0,ScalingAdjustment=-2
```

记下策略的 Amazon Resource Name (ARN)。创建 CloudWatch 警报时，您需要 ARN。

示例 6：为向外扩展应用简单扩展策略

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-out-policy \
  --auto-scaling-group-name my-asg --scaling-adjustment 30 \
  --adjustment-type PercentChangeInCapacity --min-adjustment-magnitude 2
```

记下策略的 Amazon Resource Name (ARN)。创建 CloudWatch 警报时，您需要 ARN。

示例 7：为缩减应用简单扩展策略

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-in-policy \
  --auto-scaling-group-name my-asg --scaling-adjustment -1 \
  --adjustment-type ChangeInCapacity --cooldown 180
```

记下策略的 Amazon Resource Name (ARN)。创建 CloudWatch 警报时，您需要 ARN。

Amazon EC2 Auto Scaling 的预测式扩展

使用预测式扩展可在流量流的每日和每周模式之前增加 Auto Scaling 组中的 EC2 实例数。

预测式扩展非常适合以下情况：

- 周期性流量，例如正常营业时间内的高资源利用率以及晚上和周末的低资源利用率
- 重复的打开和关闭工作负载模式，例如批处理、测试或定期数据分析
- 初始化需要很长时间的应用程序，从而在向外扩展事件期间对应用程序性能造成明显的延迟影响

一般来说，如果您有定期的流量增长模式以及需要很长时间才能初始化的应用程序，则应考虑使用预测式扩展。与仅使用动态扩展相比，预测式扩展可以通过在预测负载之前启动容量来帮助您更快地扩展。预测式扩展还可以帮助您避免需要过度配置容量，从而节省 EC2 账单上的资金。

例如，考虑在营业时间内具有高利用率以及夜间具有低利用率的应用程序。在每个工作日开始时，预测式扩展可以在流量第一次涌入之前增加容量。这有助于您的应用程序在利用率较低的时期内保持高可用性和性能。您不必等待动态扩展来响应不断变化的流量。您也不必花时间查看应用程序的负载模式，并尝试使用计划扩展计划适当的容量。

使用 Amazon Web Services Management Console、Amazon CLI 或其中一个 SDK，将预测式扩展策略添加到任何 Auto Scaling 组。

目录

- 预测式扩展的工作方式 (p. 166)
- 最佳实践 (p. 167)
- 创建预测式扩展策略（控制台） (p. 167)
- 创建预测式扩展策略 (Amazon CLI) (p. 170)
- 限制 (p. 172)
- 支持的区域 (p. 172)
- 评估预测性扩展策略 (p. 172)
- 使用计划操作覆盖预测值 (p. 177)
- 使用自定义指标进行高级预测性扩展策略配置 (p. 180)

预测式扩展的工作方式

预测式扩展使用机器学习来根据 CloudWatch 的历史数据预测容量需求。机器学习算法会消耗可用的历史数据并计算最适合历史负载模式的容量，然后根据新数据不断学习，以使未来的预测更加准确。

要使用预测式扩展，首先需要创建包含一对指标和目标利用率的扩展策略。如果至少有 24 小时的历史数据，则在创建策略后立即开始创建预测。预测式扩展查找前 14 天的 CloudWatch 指标数据中的模式，以创建未来 48 小时的每小时预测。根据最新的 CloudWatch 指标数据每六小时更新一次预测数据。

您可以在仅预测模式下配置预测式扩展，以便您可以在预测式扩展开始主动扩展容量之前对预测进行评估。然后，您可以从 Amazon EC2 Auto Scaling 控制台以图表形式查看 CloudWatch 中的预测和最近指标数据。您还可以通过使用 Amazon CLI 或其中一个 SDK 来访问预测数据。

当您准备好使用预测式扩展开始扩展时，请将策略从仅预测模式切换到预测和扩展模式。切换到预测和扩展模式时，Auto Scaling 组将根据预测开始扩展。

通过使用预测，Amazon EC2 Auto Scaling 可在每小时开始时扩展实例数：

- 如果实际容量小于预测容量，Amazon EC2 Auto Scaling 会向外扩展您的 Auto Scaling 组，使其所需容量与预测容量相等。
- 如果实际容量大于预测容量，则 Amazon EC2 Auto Scaling 不会缩小容量。
- 如果预测容量超出此范围，则会遵守为 Auto Scaling 组的最小容量和最大容量设置的值。

最佳实践

- 确认预测式扩展是否适合您的工作负载。如果工作负载显示特定于星期几的什么时间的重复负载模式，则工作负载非常适合预测式扩展。若要检查这一点，请在仅预测模式下配置预测性扩展策略，然后参考控制台中的建议。Amazon EC2 Auto Scaling 会根据对潜在策略性能的观察提供建议。在允许预测式扩展主动扩展应用程序之前，请评估预测及建议。
- 预测式扩展至少需要 24 小时的历史数据才能开始预测。但是，如果历史数据跨越整整两周，预测会更有效。如果您通过创建新的 Auto Scaling 组并删除旧组来更新应用程序，则新的 Auto Scaling 组需要 24 小时的历史加载数据，然后预测式扩展才能再次开始生成预测。您可以使用自定义指标来聚合新旧自动扩缩组之间的指标。否则，您可能需要等待几天才能获得更准确的预测。
- 首先，使用 Amazon EC2 Auto Scaling 控制台在仅预测模式下创建多个预测性扩展策略。这会测试不同指标和目标值的潜在影响。您可以为每个 Auto Scaling 组创建多个预测式扩展策略，但只能将其中一个策略用于主动扩展。
- 在选择负载指标时，请确保其数据描述应用程序的全部负载。还要确保它与您想要扩展的性能方面相关。
- 使用具有动态扩展功能的预测式扩展。动态扩展用于根据资源利用率的实时变化自动扩展容量。将其与预测式扩展结合使用可帮助您紧密跟踪应用程序的需求曲线，在低流量期间进行扩展并在流量高于预期时向外扩展。当多个扩展策略处于活动状态时，每个策略将独立确定所需容量，并将所需容量设置为其中的最大容量。例如，如果要求 10 个实例保持目标跟踪扩展策略中的目标利用率，并且需要 8 个实例保持在预测式扩展策略中的目标利用率，则组的所需容量设置为 10。对于大多数想要开始使用动态扩展的客户而言，我们建议使用目标跟踪扩展策略。有关更多信息，请参阅Amazon EC2 Auto Scaling 的目标跟踪扩缩策略 (p. 133)。

创建预测式扩展策略（控制台）

您可以通过 Amazon EC2 Auto Scaling 控制台创建、查看和删除您的预测性扩缩策略。

通过控制台创建预测性扩缩策略（预定义指标）

按照以下程序，使用预定义的指标（每个目标的 CPU、网络 I/O 或应用程序负载均衡器请求数）创建预测性扩缩策略。创建预测性扩缩策略的最简单方式，是使用预定义指标。如果您希望使用自定义指标，请参阅 通过控制台创建预测性扩缩策略（自定义指标）(p. 168)。

如果自动扩缩组为新组，则必须提供至少 24 小时的数据，然后 Amazon EC2 Auto Scaling 才会为其生成预测。

创建预测式扩展策略

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在页面底部打开一个拆分窗格。
3. 在自动扩展选项卡的扩展策略中，选择创建预测式扩展策略。
4. 输入策略的名称。
5. 启用根据预测进行扩展，授予 Amazon EC2 Auto Scaling 立即开始扩展的权限。

   若要将策略保持在仅预测模式，请保持根据预测进行扩展关闭状态。
6. 对于指标，从选项列表中选择指标。选项包括 CPU、网络输入、网络输出、Application Load Balancer 请求计数和自定义指标对。

   如果您选择每目标的 Application Load Balancer 请求计数，请在目标组中选择目标组。每目标的 Application Load Balancer 请求计数仅在您已将 Application Load Balancer 目标组附加到 Auto Scaling 组时才支持。

   如果您选择自定义指标对，请从负载指标和扩展指标的下拉列表中选择单个指标。
7. 对于目标利用率，请输入 Amazon EC2 Auto Scaling 应维护的目标值。Amazon EC2 Auto Scaling 可向外扩展您的容量直到平均利用率达到目标利用率，或直到达到您指定的最大实例数。

如果您的扩展指标是...	然后目标利用率表示...
CPU	每个实例理想情况下应使用的 CPU 百分比。
网络输入	每个实例理想情况下应接收的平均每分钟字节数。
网络输出	每个实例理想情况下应发出的平均每分钟字节数。
每目标的 Application Load Balancer 请求计数	每个实例理想情况下应接收的平均每分钟请求数。

8. （可选）对于预启动实例，请选择您希望在预测调用增加负载之前启动实例的距离。
9. （可选）对于 Max capacity behavior（最大容量行为），请选择当预测容量超过定义的最大值时，是否允许 Amazon EC2 Auto Scaling 横向扩展至高于该组最大容量的水平。通过开启此设置，您将可以在预测您的流量会触及最大值期间进行横向扩展。
10. （可选）对于高于预测容量的缓冲区最大容量，选择在预测容量接近或超过最大容量时要使用的附加容量。该值是作为相对于预测容量的百分比指定的。例如，如果缓冲区为 10，这意味着 10% 的缓冲区。因此，如果预测容量为 50，最大容量为 40，则有效的最大容量是 55。

    如果设置为 0，Amazon EC2 Auto Scaling 可以将容量扩展到高于最大容量，直至等于但不能超过预测容量的水平。
11. 选择创建预测式扩展策略。

通过控制台创建预测性扩缩策略（自定义指标）

按照以下程序，使用自定义指标创建预测性扩缩策略。自定义指标可以包括 CloudWatch 提供的其他指标以及您发布到 CloudWatch 的指标。要使用每个目标的 CPU、网络 I/O 或应用程序负载均衡器请求数指标，请参阅 通过控制台创建预测性扩缩策略（预定义指标）(p. 167)。

要使用自定义指标创建预测性扩缩策略，您必须执行以下操作：

- 您必须提供可让 Amazon EC2 Auto Scaling 与 CloudWatch 中的指标进行交互的原始查询。有关更多信息，请参阅使用自定义指标进行高级预测性扩展策略配置 (p. 180)。为确保 Amazon EC2 Auto Scaling

可以从 CloudWatch 提取指标数据，请确认每个查询都返回了数据点。您可以使用 CloudWatch 控制台或 CloudWatch GetMetricData API 操作来确认这一点。

Note

我们在 Amazon EC2 Auto Scaling 控制台中的 JSON 编辑器中提供了示例 JSON 负载。这些示例提供了添加由 Amazon 提供的其他 CloudWatch 指标或您之前发布到 CloudWatch 的指标所需键值对的参考。您可以将这些示例负载作为起点，然后根据需要进行自定义。

- 如果您使用任何指标数学，则必须手动构造适合您的独特应用场景的 JSON。有关更多信息，请参阅使用指标数学表达式 (p. 182)。在策略中使用指标数学之前，请确认基于指标数学表达式的指标查询有效并返回了单个时间序列。您可以使用 CloudWatch 控制台或 CloudWatch GetMetricData API 操作来确认这一点。

如果因提供的数据错误（例如自动扩缩组名称出错），您的查询出现错误，则预测将没有任何数据。要排查自定义指标问题，请参阅 注意事项和问题排查 (p. 186)。

创建预测式扩展策略

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在页面底部打开一个拆分窗格。
3. 在自动扩展选项卡的扩展策略中，选择创建预测式扩展策略。
4. 输入策略的名称。
5. 启用根据预测进行扩展，授予 Amazon EC2 Auto Scaling 立即开始扩展的权限。

   若要将策略保持在仅预测模式，请保持根据预测进行扩展关闭状态。
6. 对于 Metrics（指标），选择 Custom metric pair（自定义指标对）。

   a. 对于 Load metric（负载指标），选择 Custom CloudWatch metric（自定义 CloudWatch 指标）以使用自定义指标。构造包含策略的负载指标定义的 JSON 负载，然后将其粘贴到 JSON 编辑器框中，替换其中已有的内容。

   b. 对于 Scaling metric（扩缩指标），选择 Custom CloudWatch metric（自定义 CloudWatch 指标）以使用自定义指标。构造包含策略的扩缩指标定义的 JSON 负载，然后将其粘贴到 JSON 编辑器框中，替换其中已有的内容。

   c. （可选）要添加自定义容量指标，请选中 Add custom capacity metric（添加自定义容量指标）复选框。构造包含策略的容量指标定义的 JSON 负载，然后将其粘贴到 JSON 编辑器框中，替换其中已有的内容。

      如果您的容量指标数据跨越多个自动扩缩组，则只需启用此选项即可创建新的容量时间序列。在这种情况下，必须使用指标数学将数据聚合成单个时间序列。
7. 对于目标利用率，请输入 Amazon EC2 Auto Scaling 应维护的目标值。Amazon EC2 Auto Scaling 可向外扩展您的容量直到平均利用率达到目标利用率，或直到达到您指定的最大实例数。
8. （可选）对于预启动实例，请选择您希望在预测调用增加负载之前启动实例的距离。
9. （可选）对于 Max capacity behavior（最大容量行为），请选择当预测容量超过定义的最大值时，是否允许 Amazon EC2 Auto Scaling 横向扩展至高于该组最大容量的水平。通过开启此设置，您将可以在预测您的流量会触及最大值期间进行横向扩展。
10. （可选）对于高于预测容量的缓冲区最大容量，选择在预测容量接近或超过最大容量时要使用的附加容量。该值是作为相对于预测容量的百分比指定的。例如，如果缓冲区为 10，这意味着 10% 的缓冲区。因此，如果预测容量为 50，最大容量为 40，则有效的最大容量是 55。

    如果设置为 0，Amazon EC2 Auto Scaling 可以将容量扩展到高于最大容量，直至等于但不能超过预测容量的水平。
11. 选择创建预测式扩展策略。

创建预测式扩展策略 (Amazon CLI)

使用以下 Amazon CLI 为 Auto Scaling 组配置预测式扩展策略。有关可以为预测式扩展策略指定 CloudWatch 指标的更多信息，请参阅 Amazon EC2 Auto Scaling API 参考中的 PredictiveScalingMetricSpecification。

示例 1：创建预测但不扩展的预测式扩展策略

以下示例策略显示了完整的策略配置，该配置使用 CPU 利用率指标进行预测式扩展，其中目标利用率为 40。默认使用 ForecastOnly 模式，除非您明确指定要使用的模式。将此配置保存在名为 config.json 的文件中。

```
{
    "MetricSpecifications": [
        {
            "TargetValue": 40,
            "PredefinedMetricPairSpecification": {
                "PredefinedMetricType": "ASGCPUUtilization"
            }
        }
    ]
}
```

要通过命令行创建此策略，请运行 put-scaling-policy 命令并指定此配置文件，如下例所示。

```
aws autoscaling put-scaling-policy --policy-name cpu40-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

如果成功，此命令将返回策略的 Amazon Resource Name (ARN)。

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/cpu40-predictive-scaling-policy",
  "Alarms": []
}
```

示例 2：预测和扩展的预测式扩展策略

对于允许 Amazon EC2 Auto Scaling 预测和扩展的策略，请添加值为 ForecastAndScale 的属性 Mode。以下示例显示了使用 Application Load Balancer 请求计数指标的策略配置。目标利用率是 1000，并且预测式扩展设置为 ForecastAndScale 模式。

```
{
    "MetricSpecifications": [
        {
            "TargetValue": 1000,
            "PredefinedMetricPairSpecification": {
                "PredefinedMetricType": "ALBRequestCount",
                "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-
group/943f017f100becff"
            }
        }
    ],
    "Mode": "ForecastAndScale"
}
```

要创建此策略，请运行指定配置文件的 put-scaling-policy 命令，如下例所示。

```
aws autoscaling put-scaling-policy --policy-name alb1000-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

如果成功，此命令将返回策略的 Amazon Resource Name (ARN)。

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-
id:scalingPolicy:19556d63-7914-4997-8c81-d27ca5241386:autoScalingGroupName/my-
asg:policyName/alb1000-predictive-scaling-policy",
  "Alarms": []
}
```

示例 3：可扩展大于最大容量的预测式扩展策略

以下示例显示如何创建一个策略，该策略可以在您需要它来处理高于正常负载时扩展高于组的最大大小限制。预设情况下，Amazon EC2 Auto Scaling 的 EC2 扩展容量不会超过您定义的最大容量。但是，如果让它扩展更高，容量稍微增加以避免性能或可用性问题，可能会有所帮助。

要为 Amazon EC2 Auto Scaling 提供空间，以便在容量预计达到或非常接近组的最大大小时配置额外容量，请指定 `MaxCapacityBreachBehavior` 和 `MaxCapacityBuffer` 属性，如以下示例所示。您必须指定值为 `IncreaseMaxCapacity` 的 `MaxCapacityBreachBehavior`。您的组可以具有的最大实例数取决于 `MaxCapacityBuffer` 值。

```
{
    "MetricSpecifications": [
        {
            "TargetValue": 70,
            "PredefinedMetricPairSpecification": {
                "PredefinedMetricType": "ASGCPUUtilization"
            }
        }
    ],
    "MaxCapacityBreachBehavior": "IncreaseMaxCapacity",
    "MaxCapacityBuffer": 10
}
```

在本示例中，该策略配置为使用 10% 缓冲区 (`"MaxCapacityBuffer": 10`)，因此如果预测容量为 50 并且最大容量为 40，则实际的最大容量为 55。如果策略可以将容量扩展到高于最大容量以等于但不超过预测容量，则缓冲区为 0 (`"MaxCapacityBuffer": 0`)。

要创建此策略，请运行指定配置文件的 put-scaling-policy 命令，如下例所示。

```
aws autoscaling put-scaling-policy --policy-name cpu70-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

如果成功，此命令将返回策略的 Amazon Resource Name (ARN)。

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:d02ef525-8651-4314-
bf14-888331ebd04f:autoScalingGroupName/my-asg:policyName/cpu70-predictive-scaling-policy",
  "Alarms": []
}
```

限制

- 预测式扩展需要 24 小时的指标历史记录才能生成预测。
- 预测性扩展的核心假设是 Auto Scaling 组是同质的，且所有实例的容量相等。如果您的组不是这样，预测的容量可能不准确。因此，在为混合实例组 (p. 50)创建预测性扩缩策略时务必谨慎，因为可以调配容量不相等的不同类型的实例。以下是一些预测容量不准确的例子：
  - 您的预测性扩缩策略基于 CPU 利用率，但每个 Auto Scaling 实例上的 vCPU 数量因实例类型而异。
  - 您的预测性扩缩策略基于网络进入或网络外出，但每个 Auto Scaling 实例的网络带宽吞吐量因实例类型而异。例如，M5 和 M5n 实例类型相似，但 M5n 实例类型显著提高了网络吞吐量。

支持的区域

Amazon EC2 Auto Scaling 在下列 Amazon Web Services 区域支持预测性扩缩策略：美国东部（弗吉尼亚州北部）、美国东部（俄亥俄州）、美国西部（俄勒冈州）、美国西部（北加利福尼亚）、非洲（开普敦）、加拿大（中部）、欧洲地区（法兰克福）、欧洲地区（爱尔兰）、欧洲地区（伦敦）、欧洲地区（米兰）、欧洲地区（巴黎）、欧洲地区（斯德哥尔摩）、亚太地区（香港）、亚太地区（雅加达）、亚太地区（孟买）、亚太地区（大阪）、亚太地区（东京)、亚太地区（新加坡）、亚太地区（首尔）、亚太地区（悉尼）、中东（巴林）、南美洲（圣保罗）、中国（北京）、中国（宁夏）和 Amazon GovCloud（美国西部）。

评估预测性扩展策略

在使用预测性扩展策略扩展您的自动扩缩组之前，请在 Amazon EC2 Auto Scaling 控制台中查看您策略的建议和其他数据。这很重要，因为在确定预测准确之前，您不希望使用预测扩展策略来扩展实际容量。

如果自动扩缩组是新的，则需要 24 小时的时间让 Amazon EC2 Auto Scaling 创建第一个预测。

当 Amazon EC2 Auto Scaling 创建预测时，它使用历史数据。如果您的自动扩缩组还没有太多最近的历史数据，Amazon EC2 Auto Scaling 可能会使用从当前可用的历史聚合中创建的聚合数据临时回填预测。预测会在策略创建日期前最多回填两周。

目录

- 查看您的预测性扩展建议 (p. 172)
- 查看预测性扩展监控图表 (p. 173)
- 使用 CloudWatch 监控预测性扩缩指标 (p. 174)

查看您的预测性扩展建议

为了进行有效的分析，Amazon EC2 Auto Scaling 应至少有两个预测性扩展策略可供比较。（但您仍可以查看单个策略的调查结果。）创建多个策略时，您可以对使用一个指标的策略和使用另一个指标的策略进行评估。您还可以评估不同目标值和指标组合的影响。创建预测性扩展策略后，Amazon EC2 Auto Scaling 会立即开始评估哪种策略可以更好地扩展您的组。

在 Amazon EC2 Auto Scaling 控制台中查看您的建议

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在页面底部打开一个拆分窗格。
3. 在 Auto Scaling 选项卡的预测性扩展策略下，您可以查看有关策略的详细信息以及我们的建议。该建议告诉您预测性扩展策略是否比不使用预测性扩展策略做得更好。

如果您不确定预测性扩展策略是否适合您的组，请查看可用性影响和成本影响列以选择正确的策略。每列的信息告诉您该策略的影响。

- 可用性影响：描述与不使用策略相比，该策略是否可以通过预置足够的实例来处理工作负载，从而避免对可用性的负面影响。
- 成本影响：描述与不使用策略相比，该策略是否可以通过不过度预置实例而避免对您的成本产生负面影响。过度预置会导致您的实例未得到充分利用或处于闲置状态，这只会增加对成本的影响。

如果您有多个策略，则在以较低成本提供最大可用性优势的策略名称旁边显示最佳预测标签。对可用性的影响给予了更多的重视。

4. （可选）要为建议结果选择所需的时间段，请从评估期下拉列表中选择您的首选值：2 天、1 周、2 周、4 周、6 周或 8 周。默认情况下，评估期为过去两周。更长的评估期可为建议结果提供更多的数据点。但是，如果您的负载模式发生了变化，例如在需求异常之后，添加更多数据点可能不会改善结果。在这种情况下，您可以通过查看最新数据来获得更有针对性的建议。

   Note

   仅为处于仅预测模式的策略生成建议。当策略在整个评估期内处于仅预测模式时，建议功能的效果会更好。如果您在预测和扩展模式下启动策略，稍后将其切换为仅预测模式，则该策略的调查结果可能会有偏差。这是因为该策略已经为实际容量做出了贡献。

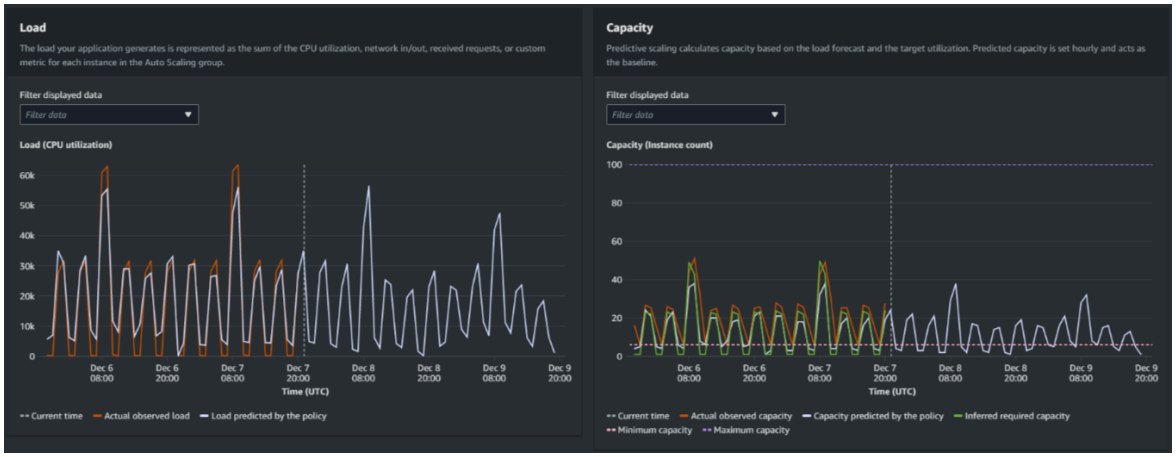
查看预测性扩展监控图表

在 Amazon EC2 Auto Scaling 控制台中，您可以查看前几天、几周或几个月的预测，以便可视化策略在一段时间内的表现如何。在决定是否让策略扩展您的实际容量时，您还可以使用这些信息来评估预测的准确性。

在 Amazon EC2 Auto Scaling 控制台中查看预测性扩展监控图表

1. 从预测性扩展策略列表中选择一个策略。
2. 在监控部分中，您可以根据实际值查看策略对过去和未来负载和容量的预测。负载图表显示所选负载指标的负载预测和实际值。容量图表显示策略预测的实例数量。它还包括启动实例的实际数量。垂直线将历史值与未来预测分开。创建策略后不久，这些图表可用。
3. （可选）要更改图表中显示的历史数据量，请从页面顶部的评估期下拉列表中选择您的首选值。评估期不会以任何方式转换此页面上的数据。它只会更改显示的历史数据量。

下图显示了多次应用预测时的负载和容量图表。预测性扩展会根据您的历史负载数据来预测负载。应用程序生成的负载由自动扩缩组中每个实例的 CPU 利用率、网络输入/输出、收到的请求或自定义指标的总和来表示。预测性扩展会根据负载预测和您希望扩展指标达到的目标利用率来计算未来的容量需求。



比较负载图表中的数据

每条水平线代表一小时间隔内报告的一组不同的数据点：

1. 实际观测负载使用所选负载指标的 SUM 统计数据来显示过去每小时的总负载。
2. 策略预测的负载显示每小时的负载预测。该预测基于前两周的实际负载观测结果。

比较容量图表中的数据

每条水平线代表一小时间隔内报告的一组不同的数据点：

1. 实际观测容量显示您的自动扩缩组过去的实际容量，这取决于您的其他扩展策略和所选时间段内有效的最小组大小。
2. 策略预测的容量显示当策略处于预测和扩展模式时，您在每个小时开始时期望拥有的基准容量。
3. 推断的所需容量显示将扩展指标维持在您所选择目标值的理想容量。
4. 最小容量显示自动扩缩组的最小容量。
5. 最大容量显示自动扩缩组的最大容量。

为了计算推断的所需容量，我们首先假设每个实例在指定目标值下的利用率相等。实际上，实例的利用率并不均等。但是，通过假设实例之间的利用率分布均匀，我们可以对所需的容量进行可能的估计。然后计算容量需求与您在预测性扩展策略中使用的扩展指标成反比。换句话说，随着容量的增加，扩展指标会以相同的速率减少。例如，如果容量翻倍，则扩展指标必定会减少一半。

推断的所需容量的公式：

```
sum of (actualCapacityUnits*scalingMetricValue)/(targetUtilization)
```

例如，我们在给定一小时内使用 `actualCapacityUnits`（`10`）和 `scalingMetricValue`（`30`）。然后，我们采用您在预测扩展策略（`60`）中指定的 `targetUtilization`，计算同一小时的推断所需容量。这会返回值 5。这意味着 5 是推断出的维持容量所需的容量，与扩展指标的目标值成反比。

> **Note**
>
> 您可以使用各种杠杆来调整和提高应用程序的成本节约和可用性。
>
> - 您可以使用预测性扩展来计算基准容量，使用动态扩展来处理额外的容量。动态扩展独立于预测性扩展，会根据当前的利用率向内和向外扩展。首先，Amazon EC2 Auto Scaling 计算每个动态扩展策略的建议实例数。然后，它会根据提供最多实例的策略进行扩展。
> - 要允许在负载减少时进行横向缩减，您的自动扩缩组应始终至少有一个启用了横向缩减部分的动态扩展策略。
> - 您可以通过确保最小和最大容量不太严格来提高扩展性能。如果策略中包含的推荐实例数不在最小和最大容量范围内，则将阻止横向缩减和横向扩展。

使用 CloudWatch 监控预测性扩缩指标

根据您的需求，您可能更适合从 Amazon CloudWatch 而非 Amazon EC2 Auto Scaling 控制台访问预测性扩缩监控数据。创建预测性扩缩策略后，该策略将收集用于预测未来负载和容量的数据。收集这些数据后，系统会自动按固定间隔时间将其存储在 CloudWatch 中。然后，您可以使用 CloudWatch 直观地显示随时间变化的策略性能。您还可以创建 CloudWatch 警报，以便在性能指标的变化超出您在 CloudWatch 中定义的限制时通知您。

主题

- 可视化显示历史预测数据 (p. 175)

- 使用指标数学创建准确度指标 (p. 175)

可视化显示历史预测数据

您可以在 CloudWatch 中查看预测性扩缩策略的负载和容量预测数据。它可以在单个图表中直观地显示相比其他 CloudWatch 指标的预测，因此非常有用。您还可以查看更大的时间范围，以了解长期的趋势，这也非常有益。您可以访问长达 15 个月的历史指标，以更好地了解您的策略性能。

有关更多信息，请参阅预测性扩缩指标和维度 (p. 251)。

使用 CloudWatch 控制台查看历史预测数据

1. 通过以下网址打开 CloudWatch 控制台：https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，选择 Metrics（指标），然后选择 All metrics（所有指标）。
3. 选择 Auto Scaling（自动扩缩）指标命名空间。
4. 选择下面的一个选项，以查看负载预测或容量预测指标：
   - Predictive Scaling Load Forecasts（预测性扩缩负载预测）
   - Predictive Scaling Capacity Forecasts（预测性扩缩容量预测）
5. 在搜索字段中，输入预测性扩缩策略的名称或自动扩缩组的名称，然后按 Enter 键以筛选结果。
6. 要为指标绘制图表，请选中该指标旁的复选框。要更改图表的名称，请选择铅笔图标。要更改时间范围，请选择某个预定义的值或选择 custom。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的绘制指标图表。
7. 要更改统计数据，请选择 Graphed metrics（已绘制图表指标）选项卡。选择列标题或单个值，然后选择其他统计数据。尽管您可以为每个指标选择任何统计数据，但并非所有的统计数据都对 PredictiveScalingLoadForecast 和 PredictiveScalingCapacityForecast 指标有用。例如，Average（平均）、Minimum（最小）和 Maximum（最大）统计数据非常有用，但 Sum（总和）统计数据用处不大。
8. 要在图表中添加其他指标，请在 All（全部）下选择 Browse（浏览），找到特定的指标，然后选中它旁边的复选框。您最多可以添加 10 个指标。

   例如，要将 CPU 利用率的实际值添加到图表中，请选择 EC2 命名空间，然后选择 By Auto Scaling Group（按自动扩缩组）。然后，选中 CPUUtilization 指标和特定自动扩缩组旁的复选框。
9. （可选）要将此图表添加到 CloudWatch 控制面板，请选择 Actions（操作），然后选择 Add to dashboard（添加到控制面板）。

使用指标数学创建准确度指标

借助指标数学，您可以查询多个 CloudWatch 指标并使用数学表达式来创建基于这些指标的新时间序列。您可以在 CloudWatch 控制台上直观显示生成的时间序列，并将其添加到控制面板中。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用指标数学。

借助指标数学，您能够以不同方式绘制 Amazon EC2 Auto Scaling 为预测性扩缩而生成的数据。这可帮助您监控随时间变化的策略性能，并帮助您了解是否可以改进指标组合。

例如，您可以使用指标数学表达式来监控平均绝对百分比误差（MAPE）。MAPE 指标可帮助监控在给定预测时段内预测值与实际观测值之间的差异。MAPE 值的变化可能表明随着应用性质的变化，策略的性能是否会随着时间的推移而下降。MAPE 增加说明着预测值和实际值之间的差异加大。

示例：指标数学表达式

要开始使用此类图表，您可以创建一个与下例中类似的指标数学表达式。

```
{
  "MetricDataQueries": [
```

```
    {
      "Expression": "TIME_SERIES(AVG(ABS(m1-m2)/m1))",
      "Id": "e1",
      "Period": 3600,
      "Label": "MeanAbsolutePercentageError",
      "ReturnData": true
    },
    {
      "Id": "m1",
      "Label": "ActualLoadValues",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "CPUUtilization",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg"
            }
          ]
        },
        "Period": 3600,
        "Stat": "Sum"
      },
      "ReturnData": false
    },
    {
      "Id": "m2",
      "Label": "ForecastedLoadValues",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/AutoScaling",
          "MetricName": "PredictiveScalingLoadForecast",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg"
            },
            {
              "Name": "PolicyName",
              "Value": "my-predictive-scaling-policy"
            },
            {
              "Name": "PairIndex",
              "Value": "0"
            }
          ]
        },
        "Period": 3600,
        "Stat": "Average"
      },
      "ReturnData": false
    }
  ]
}
```

这不是单个指标，而是一组针对 MetricDataQueries 的指标数据查询结构。MetricDataQueries 中的每一项都会获取一个指标或执行一个数学表达式。第一项 e1 是一个数学表达式。指定的表达式将 ReturnData 参数设置为 true，这最终会生成单个时间序列。对于所有其他指标，ReturnData 值为 false。

在此例中，指定的表达式使用实际值和预测值作为输入，并返回新指标（MAPE）。m1 是包含实际负载值的 CloudWatch 指标（假设 CPU 利用率是最初为名为 my-predictive-scaling-policy 的策略指定的负载指标）。m2 是包含预测负载值的 CloudWatch 指标。MAPE 指标的数学语法如下所示：

(((实际值 - 预测值)/(实际值)) 的绝对值) 的平均值

可视化显示准确度指标并设置警报

要可视化显示准确度指标数据，请在 CloudWatch 控制台中选择 Metrics（指标）选项卡。您可以在此处绘制数据图表。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的将数学表达式添加到 CloudWatch 图表。

您还可以在 Metrics（指标）部分为您监控的指标设置警报。在 Graphed metrics（绘制的指标）选项卡中，选择 Actions（操作）列下的 Create alarm（创建警报）。Create alarm（创建警报）图标用一个小铃铛表示。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的创建基于指标数学表达式的 CloudWatch 警报。有关使用 Amazon SNS 接收提醒的更多信息，请参阅《Amazon CloudWatch 用户指南》中的设置 Amazon SNS 通知。

您也可以使用 GetMetricData 和 PutMetricAlarm 以使用指标数学执行计算，并基于输出创建警报。

使用计划操作覆盖预测值

有时，您可能会获得有关未来应用程序需求的其他信息，预测计算无法考虑这些信息。例如，预测计算可能会低估即将举行的市场营销活动所需的容量。您可以使用计划操作在未来时段内临时覆盖预测。计划操作可以循环运行，也可以在出现一次性需求波动的特定日期和时间运行。

例如，您可以创建具有高于预测容量的最小容量的计划操作。运行时，Amazon EC2 Auto Scaling 会更新 Auto Scaling 组的最小容量。由于预测式扩展可针对容量进行优化，因此执行最小容量高于预测值的计划操作。这样可以防止容量低于预期。要停止覆盖预测，请使用第二个计划操作将最小容量恢复到其原始设置。

以下过程概述了在将来期间覆盖预测的步骤。

目录

- 步骤 1：（可选）分析时间序列数据 (p. 177)
- 步骤 2：创建两个计划操作 (p. 179)

步骤 1：（可选）分析时间序列数据

首先分析预测时间序列数据。这是一个可选步骤，但如果您想了解预测的详细信息，它会很有帮助。

1. 检索预测

   创建预测后，您可以查询预测中的特定时间段。查询的目的是获得特定时间段的时间序列数据的完整视图。

   您的查询最多可以包含两天的未来预测数据。如果您已经使用了一段时间预测式扩展，您还可以访问过去的预测数据。但是，开始和结束时间之间的最长持续时间为 30 天。

   若要使用 get-predictive-scaling-forecast Amazon CLI 命令获取预测，请在命令中提供以下参数：

   - 在 `--auto-scaling-group-name` 参数中输入 Auto Scaling 组的名称。
   - 在 `--policy-name` 参数中输入策略的名称。
   - 在 `--start-time` 参数中输入开始时间以仅返回在指定时间或之后的预测数据。
   - 在 `--end-time` 参数中输入结束时间以仅返回在指定时间之前的预测数据。

```
aws autoscaling get-predictive-scaling-forecast --auto-scaling-group-name my-asg \
    --policy-name cpu40-predictive-scaling-policy \
    --start-time "2021-05-19T17:00:00Z" \
```

```
    --end-time "2021-05-19T23:00:00Z"
```

如果成功，该命令将返回类似于以下示例的数据。

```
{
    "LoadForecast": [
        {
            "Timestamps": [
                "2021-05-19T17:00:00+00:00",
                "2021-05-19T18:00:00+00:00",
                "2021-05-19T19:00:00+00:00",
                "2021-05-19T20:00:00+00:00",
                "2021-05-19T21:00:00+00:00",
                "2021-05-19T22:00:00+00:00",
                "2021-05-19T23:00:00+00:00"
            ],
            "Values": [
                153.0655799339254,
                128.8288551285919,
                107.1179447150675,
                197.3601844551528,
                626.4039934516954,
                596.9441277518481,
                677.9675713779869
            ],
            "MetricSpecification": {
                "TargetValue": 40.0,
                "PredefinedMetricPairSpecification": {
                    "PredefinedMetricType": "ASGCPUUtilization"
                }
            }
        }
    ],
    "CapacityForecast": {
        "Timestamps": [
            "2021-05-19T17:00:00+00:00",
            "2021-05-19T18:00:00+00:00",
            "2021-05-19T19:00:00+00:00",
            "2021-05-19T20:00:00+00:00",
            "2021-05-19T21:00:00+00:00",
            "2021-05-19T22:00:00+00:00",
            "2021-05-19T23:00:00+00:00"
        ],
        "Values": [
            2.0,
            2.0,
            2.0,
            2.0,
            4.0,
            4.0,
            4.0
        ]
    },
    "UpdateTime": "2021-05-19T01:52:50.118000+00:00"
}
```

此响应包括两个预测：LoadForecast 和 CapacityForecast。LoadForecast 显示每小时负载预测。CapacityForecast 显示每小时处理预测负载所需的容量的预测值，同时保持 TargetValue 为 40.0（平均 CPU 利用率为 40%）。

2. 确定目标时间段

   确定应发生一次性需求波动的小时数。请记住，预测中显示的日期和时间以 UTC 为单位。

步骤 2：创建两个计划操作

接下来，在应用程序的负载高于预测负载的特定时间段内创建两个计划操作。例如，如果您的营销活动会在有限时间段内使网站的流量增加，则可计划一个一次性操作以在其启动时更新最小容量。然后，安排另一个操作，以便在事件结束时将最小容量返回到原始设置。

为一次性事件创建两个计划操作（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在页面底部打开一个拆分窗格。
3. 在 Automatic scaling（自动扩展）选项卡上的 Scheduled actions（计划操作）中，选择 Create scheduled action（创建计划操作）。
4. 填写以下计划操作设置：

   a. 为计划操作输入名称。
   b. 对于最小值，输入 Auto Scaling 组的最小新容量。最小值必须小于或等于组的最大大小。如果最小值大于该组的最大大小，则必须更新最大值。
   c. 对于 Recurrence (重复次数)，选择 Once (一次)。
   d. 对于时区，请选择时区。如果未选择任何时区，预设情况下使用 ETC/UTC。
   e. 定义特定开始时间。
5. 选择 Create（创建）。

   控制台将显示 Auto Scaling 组的计划操作。
6. 配置第二个计划操作，以在事件结束时将最小容量返回原始设置。预测式扩展只能在设置最小值小于预测值时扩展容量。

为一次性事件创建两个计划操作 (Amazon CLI)

要使用 Amazon CLI 创建计划操作，请使用 put-scheduled-update-group-action 命令。

例如，让我们定义一个时间表，在 5 月 19 日下午 5:00 时保持最少三个实例的容量，持续 8 小时。以下命令显示如何实现此方案。

第一个 put-scheduled-update-group-action 命令指示 Amazon EC2 Auto Scaling 在 2021 年 5 月 19 日 UTC 下午 5:00 更新指定 Auto Scaling 组的最小容量。

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-start \
  --auto-scaling-group-name my-asg --start-time "2021-05-19T17:00:00Z" --minimum-
capacity 3
```

第二个命令指示 Amazon EC2 Auto Scaling 将该组的最小容量设置为 2021 年 5 月 20 日 UTC 上午 1:00。

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-end \
  --auto-scaling-group-name my-asg --start-time "2021-05-20T01:00:00Z" --minimum-
capacity 1
```

将这些计划操作添加到 Auto Scaling 组后，Amazon EC2 Auto Scaling 将执行以下操作：

- 2021 年 5 月 19 日下午 5:00，第一个计划的操作将运行。如果组中当前已少于三个实例，则该组会扩展到三个实例。在此期间和接下来的八小时内，如果预测容量高于实际容量，或者如果有动态扩展策略生效，Amazon EC2 Auto Scaling 可以继续向外扩展。

- 2021 年 5 月 20 日上午 1:00，将运行第二个计划的操作。这将在事件结束时将最小容量恢复为其原始设置。

根据重复性计划进行扩展

要覆盖每周相同时间段的预测，请创建两个计划操作，并使用 cron 表达式提供时间和日期逻辑。

此 cron 表达式格式包含五个空格分隔的字段：[Minute] [Hour] [Day_of_Month] [Month_of_Year] [Day_of_Week]。字段可以包含任何允许的值，包括特殊字符。

例如，下面的 cron 表达式在每周二上午 6:30 运行操作。星号用作通配符，以匹配字段的所有值。

```
30 6 * * 2
```

另请参阅

有关如何创建、列出、编辑和删除计划操作的更多信息，请参阅 Amazon EC2 Auto Scaling 的计划扩缩 (p. 186)。

使用自定义指标进行高级预测性扩展策略配置

在预测性扩展策略中，您可以使用预定义指标或自定义指标。当预定义指标（CPU、网络 I/O 和 Application Load Balancer 请求计数）未充分描述应用程序负载时，自定义指标非常有用。

使用自定义指标创建预测性扩缩策略时，您可以指定由 Amazon 提供的其他 CloudWatch 指标，也可以指定自己定义和发布的指标。您还可以使用指标数学来聚合现有的指标并将其转换为 Amazon 会不自动跟踪的新时间序列。例如，通过计算新的总和或平均值来组合数据中的值时，该操作称为执行聚合。生成的数据称为聚合。

以下部分包含了有关如何为构造策略的 JSON 结构的最佳实践和示例。

目录

- 最佳实践 (p. 180)
- 先决条件 (p. 181)
- 构造自定义指标的 JSON (p. 181)
- 注意事项和问题排查 (p. 186)
- 限制 (p. 186)

最佳实践

以下最佳实践可帮助您更有效地使用自定义指标：

- 对于负载指标规范，最有用的指标是作为一个整体表示 Auto Scaling 组负载的指标，而不管该组的容量如何。
- 对于扩展指标规范，要扩展的最有用指标是每个实例的平均吞吐量或利用率指标。
- 扩展指标必须与容量成反比。也就是说，如果 Auto Scaling 组中的实例数量增加，则扩展指标应该减少大致相同的比例。为确保预测性扩展按预期采取行动，负载指标和扩展指标还必须彼此之间密切关联。
- 目标利用率必须与扩展指标的类型匹配。对于使用 CPU 利用率的策略配置，这是目标百分比。对于使用吞吐量（例如请求数或消息数）的策略配置，这是在任何一分钟间隔内每个实例的目标请求数或目标消息数。
- 如果未遵循这些建议，那么时间序列的预测未来值可能会不正确。要验证数据是否正确，您可以在 Amazon EC2 Auto Scaling 控制台中查看预测值。或者，在创建预测性扩展策略之后，请检查 GetPredictiveScalingForecast API 返回的 `LoadForecast` 和 `CapacityForecast` 对象。

- 我们强烈建议您在仅预测模式下配置预测式扩展，以便在预测式扩展开始主动扩展容量之前对预测进行评估。

先决条件

要将自定义指标添加到预测性扩缩策略，您必须具有 `cloudwatch:GetMetricData` 权限。

要指定自己的指标而不是 Amazon 提供的指标，您必须首先将您的指标发布到 CloudWatch。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的发布自定义指标。

如果发布自己的指标，请确保以至少五分钟的频率发布数据点。Amazon EC2 Auto Scaling 根据所需的时间长度从 CloudWatch 中检索数据点。例如，负载指标规范以每小时为指标来衡量应用程序的负载。CloudWatch 使用您已发布的指标数据在任意一小时期间提供单个数据值，其方法是将所有数据点与每个一小时内的时间戳聚合起来。

构造自定义指标的 JSON

以下部分包含了有关如何配置预测行扩缩以从 CloudWatch 查询数据的示例。配置此选项有两种不同的方法，您选择的方法会影响您为预测性扩缩策略构造 JSON 时使用的格式。使用指标数学时，JSON 格式会根据所执行的指标数学进一步变化。

1. 要创建直接从 Amazon 提供的其他 CloudWatch 指标或您发布到 CloudWatch 的指标中获取数据的策略，请参阅 包含自定义负载和扩缩指标的预测性扩缩策略示例（Amazon CLI）(p. 181)。
2. 要创建可以查询多个 CloudWatch 指标并使用数学表达式以基于这些指标创建新时间序列的策略，请参阅 使用指标数学表达式 (p. 182)。

包含自定义负载和扩缩指标的预测性扩缩策略示例（Amazon CLI）

要通过 Amazon CLI 使用自定义负载和扩缩指标创建预测性扩缩策略，请将 `--predictive-scaling-configuration` 的参数存储在名为 `config.json` 的 JSON 文件中。

您可以将以下示例中的可替换值替换为您的指标和目标利用率值，从而开始添加自定义指标。

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 50,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "MetricStat": {
              "Metric": {
                "MetricName": "MyUtilizationMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
                  {
                    "Name": "MyOptionalMetricDimensionName",
                    "Value": "MyOptionalMetricDimensionValue"
                  }
                ]
              },
              "Stat": "Average"
            }
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "MetricStat": {
```

```
            "Metric": {
              "MetricName": "MyLoadMetric",
              "Namespace": "MyNameSpace",
              "Dimensions": [
                {
                  "Name": "MyOptionalMetricDimensionName",
                  "Value": "MyOptionalMetricDimensionValue"
                }
              ]
            },
            "Stat": "Sum"
          }
        }
      ]
    }
  ]
}
```

有关更多信息，请参阅《Amazon EC2 Auto Scaling API 参考》中的 MetricDataQuery。

Note

以下是一些可以帮助您查找 CloudWatch 指标的指标名称、命名空间、维度和统计数据的其他资源：

- 有关 Amazon 服务可用指标的更多信息，请参阅 Amazon CloudWatch 用户指南中的发布 CloudWatch 指标的 Amazon 服务。
- 要获取带有 Amazon CLI 的 CloudWatch 指标的确切指标名称、命名空间和维度（如果适用），请参阅列出指标.

要创建此策略，请运行 put-scaling-policy 命令并将此 JSON 文件作为输入，如下例所示。

```
aws autoscaling put-scaling-policy --policy-name my-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

如果成功，此命令将返回策略的 Amazon Resource Name (ARN)。

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-predictive-scaling-policy",
  "Alarms": []
}
```

使用指标数学表达式

以下部分提供了预测性扩缩策略的信息和示例，这些示例演示了如何在策略中使用指标数学。

目录

- 了解指标数学 (p. 182)
- 使用指标数学组合指标的预测性扩缩策略示例（Amazon CLI）(p. 183)
- 在蓝/绿部署场景中使用的预测扩缩策略示例（Amazon CLI）(p. 184)

了解指标数学

如果您想要做的只是聚合现有指标数据，那么 CloudWatch 指标数学可以节省向 CloudWatch 发布另一个指标的精力和成本。您可以使用 Amazon 提供的任何指标，还可以使用定义为应用程序一部分的指标。例如，

您可能想要计算每个实例的 Amazon SQS 队列积压。您可以从队列中获取用于检索的可用消息的大约数量，然后将该数量除以 Auto Scaling 组的运行容量来实现这一点。

有关更多信息，请参阅 Amazon CloudWatch User Guide (Amazon CloudWatch 用户指南) 中的 Using metric math (使用指标数学)。

如果您选择在预测性扩展策略中使用指标数学表达式，请考虑以下几点：

- 指标数学运算使用指标名称、命名空间和维度键/值对指标的唯一组合的数据点。
- 您可以使用任意算术运算符 (+ - * / ^)、统计函数（例如 AVG 或 SUM）或 CloudWatch 支持的其他函数。
- 您可以在数学表达式的公式中同时使用指标和其他数学表达式的结果。
- 指标数学表达式可以由不同的聚合组成。但是，得到最终聚合结果的最佳实践是针对扩展指标使用 `Average` 以及针对负载指标使用 `Sum`。
- 指标规范中使用的任何表达式最终都必须返回一个单个时间序列。

要使用指标数学，请执行以下操作：

- 选择一个或多个 CloudWatch 指标。然后，创建表达式。有关更多信息，请参阅 Amazon CloudWatch User Guide (Amazon CloudWatch 用户指南) 中的 Using metric math (使用指标数学)。
- 使用 CloudWatch 控制台或 CloudWatch GetMetricData API 验证指标数学表达式是否有效。

使用指标数学组合指标的预测性扩缩策略示例（Amazon CLI）

有时，您可能需要首先以某种方式处理其数据，而不是直接指定指标。例如，您可能有一个从 Amazon SQS 队列中提取工作的应用程序，并且可能希望使用队列中的项目数作为预测性扩展的标准。队列中的消息数不仅仅定义您需要的实例数。因此，需要执行更多工作来创建可用于计算每个实例的积压的指标。有关更多信息，请参阅 根据 Amazon SQS 进行扩展 (p. 155)。

以下示例是适用于此场景的预测扩展策略示例。它指定了基于 Amazon SQS `ApproximateNumberOfMessagesVisible` 指标的扩展和负载指标，即可从队列中获取的用于检索的消息数量。它还使用 Amazon EC2 Auto Scaling `GroupInServiceInstances` 指标和数学表达式，计算扩展指标的每个实例的积压。

```
aws autoscaling put-scaling-policy --policy-name my-sqs-custom-metrics-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
{
  "MetricSpecifications": [
    {
      "TargetValue": 100,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Label": "Get the queue size (the number of messages waiting to be processed)",
            "Id": "queue_size",
            "MetricStat": {
              "Metric": {
                "MetricName": "ApproximateNumberOfMessagesVisible",
                "Namespace": "AWS/SQS",
                "Dimensions": [
                  {
                    "Name": "QueueName",
                    "Value": "my-queue"
                  }
                ]
              },
              "Stat": "Sum"
            },
            "ReturnData": false
```

```
          },
          {
            "Label": "Get the group size (the number of running instances)",
            "Id": "running_capacity",
            "MetricStat": {
              "Metric": {
                "MetricName": "GroupInServiceInstances",
                "Namespace": "AWS/AutoScaling",
                "Dimensions": [
                  {
                    "Name": "AutoScalingGroupName",
                    "Value": "my-asg"
                  }
                ]
              },
              "Stat": "Sum"
            },
            "ReturnData": false
          },
          {
            "Label": "Calculate the backlog per instance",
            "Id": "scaling_metric",
            "Expression": "queue_size / running_capacity",
            "ReturnData": true
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "ApproximateNumberOfMessagesVisible",
                "Namespace": "AWS/SQS",
                "Dimensions": [
                  {
                    "Name": "QueueName",
                    "Value": "my-queue"
                  }
                ],
              },
              "Stat": "Sum"
            },
            "ReturnData": true
          }
        ]
      }
    }
  ]
}
```

该示例返回策略的 ARN。

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-sqs-custom-metrics-policy",
  "Alarms": []
}
```

在蓝/绿部署场景中使用的预测扩缩策略示例（Amazon CLI）

搜索表达式提供了一个高级选项，您可以在其中查询多个 Auto Scaling 组中的指标并对其执行数学表达式。此选项对于蓝/绿部署尤其有效。

Note

蓝/绿部署是一种部署方法，您可以在其中创建两个独立但相同的 Auto Scaling 组。只有其中一个组接收生产流量。用户流量最初定向到较早的Auto Scaling 组（“蓝色”），而新组（“绿色”）用于测试和评估应用程序或服务的新版本。测试并接受新部署后，用户流量将转移到“绿色”的 Auto Scaling 组。然后，您可以在部署成功后删除“蓝色”组。

作为蓝/绿部署的一部分创建新的 Auto Scaling 组时，每个组的指标历史记录可以自动包含在预测性扩展策略中，而无需更改其指标规范。有关更多信息，请参阅 Amazon 计算博客中的将 EC2 Auto Scaling 预测性扩展策略与蓝/绿部署结合使用。

以下示例策略说明了如何执行此操作。在此示例中，策略使用 Amazon EC2 发出的 CPUUtilization 指标。它使用 Amazon EC2 Auto Scaling GroupInServiceInstances 指标和数学表达式，计算每个实例的扩展指标的值。它还指定了一个容量指标规范来获取 GroupInServiceInstances 指标。

根据指定的搜索条件，搜索表达式查找多个 Auto Scaling 组中实例的 CPUUtilization。如果您稍后创建了匹配相同搜索条件的新 Auto Scaling 组，则自动包含新 Auto Scaling 组中实例的 CPUUtilization。

```
aws autoscaling put-scaling-policy --policy-name my-blue-green-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
{
  "MetricSpecifications": [
    {
      "TargetValue": 25,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_sum",
            "Expression": "SUM(SEARCH('{AWS/EC2,AutoScalingGroupName} MetricName=
\"CPUUtilization\" ASG-myapp', 'Sum', 300))",
            "ReturnData": false
          },
          {
            "Id": "capacity_sum",
            "Expression": "SUM(SEARCH('{AWS/AutoScaling,AutoScalingGroupName} MetricName=
\"GroupInServiceInstances\" ASG-myapp', 'Average', 300))",
            "ReturnData": false
          },
          {
            "Id": "weighted_average",
            "Expression": "load_sum / capacity_sum",
            "ReturnData": true
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_sum",
            "Expression": "SUM(SEARCH('{AWS/EC2,AutoScalingGroupName} MetricName=
\"CPUUtilization\" ASG-myapp', 'Sum', 3600))"
          }
        ]
      },
      "CustomizedCapacityMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "capacity_sum",
            "Expression": "SUM(SEARCH('{AWS/AutoScaling,AutoScalingGroupName} MetricName=
\"GroupInServiceInstances\" ASG-myapp', 'Average', 300))"
          }
        ]
      }
```

```
    }
  ]
}
```

该示例返回策略的 ARN。

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-blue-green-predictive-scaling-
policy",
  "Alarms": []
}
```

注意事项和问题排查

如果在使用自定义指标时出现问题，建议您执行以下操作：

- 如果提供了错误消息，请阅读该消息并解决其报告的问题（如果可能）。
- 如果在蓝/绿部署方案中尝试使用搜索表达式时出现问题，请首先确保您了解如何创建查找部分匹配而非完全匹配的搜索表达式。此外，请检查您的查询是否只查找运行特定应用程序的 Auto Scaling 组。有关搜索表达式语法的更多信息，请参阅 Amazon CloudWatch 用户指南中的 CloudWatch 搜索表达式语法。
- 如果您没有提前验证表达式，则 put-scaling-policy 命令会在创建扩展策略时对其进行验证。但是，此命令有可能无法识别所检测错误的确切原因。要修复这些问题，请解决您在 get-metric-data 命令请求的回应中收到的错误。您还可以通过 CloudWatch 控制台对表达式进行问题排查。
- 查看控制台中的 Load (负载) 和 Capacity (容量) 图表时，Capacity (容量) 图表可能不显示任何数据。为确保图表具有完整的数据，请确保始终为 Auto Scaling 组启用组指标。有关更多信息，请参阅 启用 Auto Scaling 组指标（控制台）(p. 252)。
- 只有具备在其生命周期内于不同的 Auto Scaling 组中运行的应用程序时，容量指标规范才适用于蓝色/绿色部署。此自定义指标允许您提供多个 Auto Scaling 组的总容量。预测性扩展使用此功能在控制台中的 Capacity (容量) 图表内显示历史数据。
- 如果 `MetricDataQueries` 自行指定 SEARCH () 函数，而没有像 SUM () 这样的数学函数，则必须为 `ReturnData` 指定 `false`。原因在于搜索表达式可能返回多个时间序列，而基于表达式的指标规范仅可以返回一个时间序列。
- 搜索表达式中涉及的所有指标均应该具有相同的分辨率。

限制

- 您可以在一个指标规范中查询最多 10 个指标的数据点。
- 为满足此限制，一个表达式算作一个指标。

Amazon EC2 Auto Scaling 的计划扩缩

计划扩缩使您可以按照可预测的负载变化来设置您自己的扩缩计划。例如，假设您的 Web 应用程序的流量会在每周的星期三开始增加，并在星期四保持高流量状态，然后在星期五开始下降。您可以配置 Amazon EC2 Auto Scaling 的计划，以在星期三增加容量并在星期五减少容量。

要使用计划的缩放，请创建计划操作。计划操作会作为时间和日期的函数自动执行。创建计划操作时，您可以指定缩放活动应该在何时发生，以及缩放操作的新的所需大小、最小大小和最大大小。您可以创建仅扩展一次或按重复计划扩展的计划操作。

目录

- 注意事项 (p. 187)
- 定期安排 (p. 187)

- 创建和管理计划操作（控制台）(p. 187)
- 创建和管理计划操作 (Amazon CLI) (p. 189)
- 限制 (p. 191)

注意事项

创建计划的操作时，请记住以下内容：

- 计划操作将最小、最大和所需大小设置为由计划操作在计划操作所指定的时间指定的大小。请求可以选择只包含这些大小中的一个。例如，您可以创建仅指定所需容量的计划操作。但是，在某些情况下，您必须包括最小和最大大小，以确保您在操作中指定的新所需容量不超出这些限制。
- 预设情况下，您设置的重复计划采用协调世界时 (UTC)。您可以更改时间以符合本地时区或您的网络中其他部分的时区。如果您指定的时区遵守 Daylight Saving Time (DST)，它会自动调整 DST。
- 您可以暂时关闭 Auto Scaling 组的计划扩展，方法是暂停 `ScheduledActions` 过程。这有助于防止计划操作处于活动状态，而无需将其删除。然后，当您想要再次使用时，您可以恢复计划的扩展。有关更多信息，请参阅 暂停和恢复 Auto Scaling 组的进程 (p. 236)。
- 可以保证同组内计划操作的执行顺序正确，但不保证跨组的计划操作的执行顺序正确。
- 计划操作的执行时间一般为几秒钟。然而，该操作可能会比计划的开始时间延迟最多两分钟。因为系统将按照计划操作的顺序来执行 Auto Scaling 组内的操作，所以计划开始时间彼此接近的计划操作可能需要更长时间才能执行。

定期安排

您可以创建计划的操作，根据定期计划调整 Auto Scaling 组。

要使用 Amazon CLI 或 SDK 创建重复性的计划，请指定一个 cron 表达式和时区来描述何时重复执行该计划操作。您可以选择指定开始时间和/或结束时间的日期和时间。

若要使用 Amazon Web Services Management Console 中，指定定定期操作的重复模式、时区、开始时间和可选结束时间。所有重复模式选项都基于 cron 表达式。或者，您可以编写您自己的 Cron 表达式。

受支持的 cron 表达式格式由用空格分隔的五个字段组成：[Minute] [Hour] [Day_of_Month] [Month_of_Year] [Day_of_Week]。例如，Cron 表达式 `30 6 * * 2` 配置每周二的早上 6:30 再执行的计划操作。星号用作通配符，以匹配字段的所有值。有关 cron 表达式的其他示例，请参阅 https://crontab.guru/examples.html。有关以此格式编写您自己的 cron 表达式的信息，请参阅 Crontab。

仔细选择您的开始时间和结束时间。记住以下内容：

- 如果您指定开始时间，Amazon EC2 Auto Scaling 将在该时间执行操作，然后根据重复计划执行操作。
- 如果指定结束时间，则操作在此时间之后停止重复。在到达计划操作的结束时间后，它不会保留在您的账户中。
- 当使用 Amazon CLI 或 SDK 时，开始时间和结束时间必须设置为 UTC。

创建和管理计划操作（控制台）

使用本部分中的过程可以使用 Amazon Web Services Management Console。

如果您使用控制台创建计划操作并指定遵守夏令时 (DST) 的时区，则循环计划以及开始和结束时间都会自动调整 DST。

创建计划的操作

完成以下过程可创建计划的操作以扩展您的 Auto Scaling 组。

要为 Auto Scaling 组创建计划扩展操作。

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Automatic scaling（自动扩展）选项卡上的 Scheduled actions（计划操作）中，选择 Create scheduled action（创建计划操作）。
4. 为计划操作输入名称。
5. 适用于所需容量、最小值、最大值中，选择新的所需组大小以及新的最小和最大容量。
6. 对于 Recurrence（循环），请选择下列可用选项之一。

   - 如果您想要按循环计划进行扩展，请选择 Amazon EC2 Auto Scaling 运行计划操作的频率。
     - 如果您选择以 Every 开头的选项，则将为您创建 Cron 表达式。
     - 如果您选择 Cron，请输入 Cron 表达式，此表达式指定了执行操作的时间。
   - 如果只想缩放一次，请选择 Once（一次）。
7. 对于时区，请选择时区。默认为 `Etc/UTC`。

   Note

   列出的所有时区均来自 IANA 时区数据库。有关更多信息，请访问 https://en.wikipedia.org/wiki/List_of_tz_database_time_zones。
8. 定义日期和时间特定开始时间。

   - 如果您选择了循环计划，则开始时间将定义循环系列中第一个计划操作的运行时间。
   - 如果您选择了 Once 作为重复，则开始时间定义运行计划操作的日期和时间。
9. （可选）对于循环计划，您可以通过选择设置 End Time，然后选择一个日期和时间 End Time。
10. 选择 Create（创建）。控制台将显示 Auto Scaling 组的计划操作。

验证时间、日期和时区

要验证您的时间、日期和时区是否配置正确，在适用于 Auto Scaling 组的 Automatic scaling（弹性伸缩）选项卡上，选中 Scheduled actions（计划的操作）表中的 Start time（开始时间）、End time（结束时间）和 Time zone（时区）值。

Amazon EC2 Auto Scaling 显示您当地时间的开始时间和结束时间值，对指定日期和时间应用 UTC 偏移。UTC 偏移量是本地时间与 UTC 之间的差值，以小时和分钟为单位。时区的值显示您请求的时区，例如 `America/New_York`。

基于位置的时区（如 `America/New_York`）会根据夏令时自动调整。但是，基于 UTC 的时区（如 `Etc/UTC` 是绝对时间，不会针对 DST 进行调整。

例如对于您有一个定期计划，其时区为 `America/New_York`。第一个扩缩操作发生在 `America/New_York` DST 开始之前的时区。下一个扩缩操作发生在 `America/New_York` DST 开始之后的时区。第一个动作从当地时间 UTC-5 上午 8 点开始，而第二个时间从当地时间 UTC-4 凌晨 8 点开始。

更新计划的操作

创建计划操作后，您可以更新除名称之外的任何设置。

更新计划的操作

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。

2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Automatic scaling（自动扩展）选项卡上的 Scheduled actions（计划操作）中，选择计划操作。
4. 选择 Actions（操作）和 Edit（编辑）。
5. 进行必要更改，然后选择 Save changes（保存更改）。

删除计划的操作

当您不再需要计划操作时，可以将其删除。

删除计划操作

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选择您的 Auto Scaling 组。
3. 在 Automatic scaling（自动扩展）选项卡上的 Scheduled actions（计划操作）中，选择计划操作。
4. 依次选择 Actions（操作）和 Delete（删除）。
5. 当系统提示进行确认时，选择 Yes, Delete（是，删除）。

创建和管理计划操作 (Amazon CLI)

您可以使用 put-scheduled-update-group-action 命令，创建并更新仅扩展一次或按重复计划扩展的已计划的操作。

创建仅发生一次的计划操作

要在指定的日期和时间仅弹性伸缩可扩展 Auto Scaling 小组一次，请使用 `--start-time "YYYY-MM-DDThh:mm:ssZ"` 选项。

示例：仅向外扩展一次

要在特定时间增加 Auto Scaling 组中正在运行的实例数量，请使用以下命令。

在为 `--start-time`（世界协调时 2021 年 3 月 31 日上午 8 点）指定的日期和时间，如果组中当前包含 3 个以下实例，则该组扩展到 3 个实例。

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-one-time-action \
  --auto-scaling-group-name my-asg --start-time "2021-03-31T08:00:00Z" --desired-capacity 3
```

示例：仅向内扩展一次

要在特定时间减少 Auto Scaling 组中正在运行的实例数量，请使用以下命令。

在为 `--start-time`（世界协调时 2021 年 3 月 31 日下午 4 点）指定的日期和时间，如果组中当前包含多个实例，则该组缩减到 1 个实例。

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-one-time-action \
  --auto-scaling-group-name my-asg --start-time "2021-03-31T16:00:00Z" --desired-capacity 1
```

创建按重复计划运行的计划操作

要按重复计划来计划扩缩，请使用 `--recurrence "cron expression"` 选项。

以下是指定 cron 表达式的计划操作的示例。

在指定的计划期间（每天凌晨 9:00 (UTC)），如果组当前已少于 3 个实例，则将向外扩展至 3 个实例。如果组当前有 3 个以上的实例，它可以扩展到 3 个实例。

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-recurring-action \
  --auto-scaling-group-name my-asg --recurrence "0 9 * * *" --desired-capacity 3
```

创建指定时区的重复计划操作

默认情况下，计划操作设置为 UTC 时区。要指定不同的时区，请包含 `--time-zone` 选项并指定 IANA 时区的规范名称（例如，`America/New_York`）。有关更多信息，请访问 https://en.wikipedia.org/wiki/List_of_tz_database_time_zones。

以下是创建重复计划操作以扩展容量时使用 `--time-zone` 选项的示例。

按照指定的计划（每周一至周五本地时间下午 6:00），如果组当前具有少于 2 个实例，则它将扩展到 2 个实例。如果组当前有 2 个以上的实例，则它可以扩展到 2 个实例。

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-recurring-action \
  --auto-scaling-group-name my-asg --recurrence "0 18 * * 1-5" --time-zone "America/New_York" \
  --desired-capacity 2
```

描述计划的操作

要描述 Auto Scaling 组的计划操作，请使用以下 describe-scheduled-actions 命令。

```
aws autoscaling describe-scheduled-actions --auto-scaling-group-name my-asg
```

如果成功，该命令返回类似以下内容的输出。

```
{
  "ScheduledUpdateGroupActions": [
    {
      "AutoScalingGroupName": "my-asg",
      "ScheduledActionName": "my-recurring-action",
      "Recurrence": "30 0 1 1,6,12 *",
      "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
      "StartTime": "2020-12-01T00:30:00Z",
      "Time": "2020-12-01T00:30:00Z",
      "MinSize": 1,
      "MaxSize": 6,
      "DesiredCapacity": 4
    }
  ]
}
```

删除计划的操作

要删除计划操作，请使用以下 delete-scheduled-action 命令。

```
aws autoscaling delete-scheduled-action --auto-scaling-group-name my-asg \
  --scheduled-action-name my-recurring-action
```

限制

- 每个 Auto Scaling 组，计划操作的名称必须是唯一的。
- 计划的操作必须具有唯一时间值。如果您尝试计划在已计划另一个扩展活动的时间进行活动，则该调用将被拒绝，并返回一个错误消息，指示已存在已存在已计划启动时间的计划操作。
- 您最多可以为每个 Auto Scaling 组创建 125 个计划的操作。

Amazon EC2 Auto Scaling 生命周期钩子

Amazon EC2 Auto Scaling 能够将生命周期钩子添加到您的 Auto Scaling 组。这些钩子使 Auto Scaling 组可让您创建解决方案，这些解决方案了解 Auto Scaling 实例生命周期中的事件，然后在发生相应的生命周期事件时对实例执行自定义操作。生命周期钩子提供了指定的时间（预设情况下为 1 小时），以在实例转换到下一个状态之前等待操作完成。

作为将生命周期钩子与 Auto Scaling 实例一起使用的示例：

- 在发生向外扩展事件时，您新启动的实例将完成其启动序列并转换到等待状态。该实例处于等待状态时，它将运行脚本以下载和安装您的应用程序所需的软件包，确保您的实例在开始接收流量前已完全准备好。脚本安装完软件后，它会发送 complete-lifecycle-action 命令以继续。
- 发生缩减事件时，生命周期钩子会在实例终止前将其暂停，并使用 Amazon EventBridge 向您发送通知。当实例处于等待状态时，您可以调用 Amazon Lambda 函数或连接到实例以下载日志或其他数据，然后再将该实例完全终止。

生命周期钩子的一个常见用途是控制何时在 Elastic Load Balancing 中注册实例。通过向您的 Auto Scaling 组添加启动生命周期钩子，您可以确保引导启动脚本已成功完成，并且实例上的应用程序在生命周期钩子结束时准备好接受流量。

有关简介视频，请参阅 YouTube. 上的 Amazon 2018 重新发明：Amazon EC2 Auto Scaling 使容量管理变得轻松。

目录

- 生命周期钩子的注意事项和限制 (p. 191)
- 生命周期钩子可用性 (p. 192)
- 示例 (p. 193)
- 生命周期钩子的工作方式 (p. 193)
- 做好准备向 Auto Scaling 组添加生命周期钩子 (p. 194)
- 添加生命周期钩子 (p. 199)
- 完成生命周期操作 (p. 201)
- 教程：配置用户数据以通过实例元数据检索目标生命周期状态 (p. 202)
- 教程：配置调用 Lambda 函数的生命周期钩子 (p. 207)

生命周期钩子的注意事项和限制

使用生命周期钩子时，请记住以下注意事项和限制：

- Amazon EC2 Auto Scaling 提供了自己的生命周期，以帮助管理 Auto Scaling 组。此生命周期与其他 EC2 实例的生命周期不同。有关更多信息，请参阅Amazon EC2 Auto Scaling 实例生命周期 (p. 7)。
- 暖池中的实例也有自己的生命周期，如 暖池中实例的生命周期状态转换 (p. 217) 中所述。
- 您可以将生命周期钩子与竞价型实例一起使用，但生命周期钩子并不禁止在容量不再可用的情况下终止实例，这种情况会随时发生，并显示两分钟中断通知。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 竞价型实例中断。但是，您可以启用“Capacity Rebalancing”（容量再平衡）以主动替换已从 Amazon EC2 Spot 服务获得再平衡建议的竞价型实例，该信号是在竞价型实例处于较高中断风险时发送的。有关更多信息，请参阅使用容量重新平衡来处理 Amazon EC2 竞价型实例中断 (p. 262)。
- 横向扩展时，直到启动生命周期钩子结束之后，Amazon EC2 Auto Scaling 才会将新实例计入 Auto Scaling 组的聚合 CloudWatch 实例指标（例如 CPUUtilization、NetworkIn、NetworkOut 等）。如果未启用原定设置实例预热，或者已经启用了该功能但设置为 0，则 Auto Scaling 实例会在实例到达 `InService` 状态时立即开始向聚合实例指标提供使用情况数据。有关更多信息，请参阅为 Auto Scaling 组设置原定设置实例预热 (p. 150)。

  在横向缩减时，聚合实例指标可能无法立即反映出终止实例的移除情况。Amazon EC2 Auto Scaling 终止工作流程开始后不久，终止实例将停止作为该组聚合实例指标的一部分计入。
- 当 Auto Scaling 组启动或终止实例时，由简单扩缩策略引发的扩缩活动将暂停。如果调用生命周期钩子，由简单扩缩策略引发的扩缩活动将暂停，直至生命周期操作完成并且冷却时间已过期。为冷却时间设置较长的时间间隔意味着，恢复扩展将会需要更长的时间。有关更多信息，请参阅Amazon EC2 Auto Scaling 的缩放冷却时间 (p. 152)。
- 实例可以在有限的时间里保持等待状态。生命周期钩子的默认超时时间为一小时（检测信号超时时间）。此外，还有一个全局超时时间，它指定您可以将实例保持在等待状态的最长时间。全局超时时间为 48 小时或检测信号超时时间的 100 倍，以较小者为准。

创建生命周期钩子时，请记住以下几点：

- 您可以配置启动生命周期钩子，以便在发生意外故障时放弃启动，在这种情况下，Amazon EC2 Auto Scaling 会自动终止并替换实例。
- Amazon EC2 Auto Scaling 会限制在生命周期钩子始终失败时允许实例启动的速率，因此请务必测试并修复生命周期操作中的任何永久性错误。
- 使用 Amazon CLI、Amazon CloudFormation 或开发工具包创建和更新生命周期钩子时，会提供在从 Amazon Web Services Management Console 创建生命周期钩子时无法使用的选项。例如，用于指定 SNS 主题或 SQS 队列的 ARN 的字段不会显示在控制台中，因为 Amazon EC2 Auto Scaling 已向 Amazon EventBridge 发送事件。这些事件可以根据需要被筛选并重新导向到 Amazon 服务，如 Lambda、Amazon SNS 和 Amazon SQS。
- 您可以在创建 Auto Scaling 组时将多个生命周期钩子添加到该组，方法是使用 Amazon CLI、Amazon CloudFormation 或开发工具包调用 CreateAutoScalingGroup API。但是，如果指定，每个钩子必须具有相同的通知目标和 IAM 角色。要创建具有不同通知目标和不同角色的生命周期钩子，请在单独调用 PutLifecycleHook API 时一次性创建生命周期钩子。

生命周期钩子可用性

下表列出了可用于各种方案的生命周期钩子。

事件	实例启动或终止[1]	最大实例生命周期：替换实例	实例刷新：替换实例	容量再平衡：替换实例	温水池：进入和离开温水池的实例
实例启动	✓	✓	✓	✓	✓
实例终止	✓	✓	✓	✓	✓

[1] 适用于创建或删除组、组自动扩展或手动调整组所需容量时启动或终止的实例。当您附加或分离实例、将实例移入或移出备用模式或使用强制删除选项删除组时，不适用。

示例

我们还提供了一些 JSON 和 YAML 模板代码段，您可以用其了解如何在 Amazon CloudFormation 堆栈模板中声明生命周期钩子。有关更多信息，请参阅 Amazon CloudFormation 用户指南中的 亚马逊云科技::AutoScaling::LifecycleHook 参考。

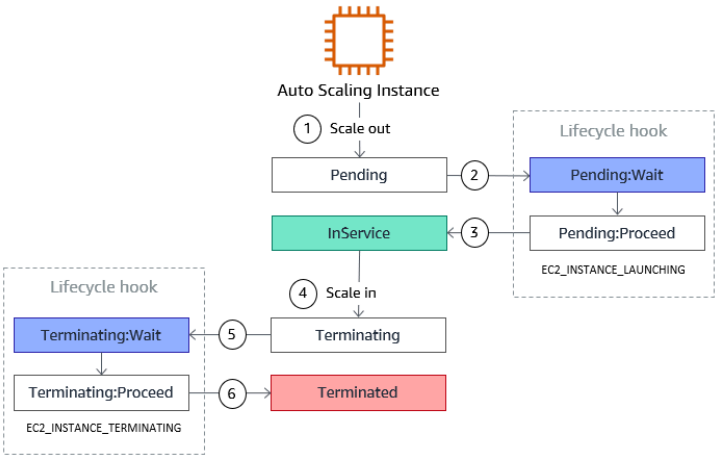
您还可以访问我们的 GitHub 存储库以下载生命周期钩子的模板和用户数据脚本。

有关使用生命周期钩子的其他示例，请参阅以下博客文章：使用 Lambda 和 Amazon EC2 运行命令为扩展实例构建备用系统和 在终止 EC2 Auto Scaling 实例之前运行代码。

生命周期钩子的工作方式

一个 Amazon EC2 实例从其启动一直到其终止，将经过不同的状态转换。您可以创建生命周期钩子，以便在实例转换到等待状态时执行操作。

下图阐释了 Auto Scaling 实例状态之间的转换。



如上图中所示：

1. Auto Scaling 组响应向外扩展事件并开始启动实例。
2. 生命周期钩子将实例置于等待状态 (`Pending:Wait`)，然后执行自定义操作。

   实例将保持等待状态，直到您完成生命周期操作，或者直到超时时段结束。预设情况下，实例将保持等待状态 1 小时，然后 Auto Scaling 组继续启动过程 (`Pending:Proceed`)。如果您需要更长时间，可通过记录检测信号来重新开始超时时段。如果您在自定义操作已完成且超时时段尚未到期时完成生命周期操作，则该时间段结束，Auto Scaling 组将继续启动过程。
3. 实例进入 `InService` 状态并开始运行状况检查宽限期。但是，在实例达到 `InService` 状态之前，如果 Auto Scaling 组与 Elastic Load Balancing 负载均衡器关联，则实例将在负载均衡器中注册，然后负载均衡器开始检查其运行状况。运行状况检查宽限期结束后，Amazon EC2 Auto Scaling 开始检查实例的运行状况。
4. Auto Scaling 组响应缩减事件并开始终止实例。如果 Auto Scaling 组与 Elastic Load Balancing 一起使用，则终止实例要首先从负载均衡器中取消注册。如果为负载均衡器启用了 Connection Draining，则实例将停止接受新连接并等待现有连接耗尽，然后再完成取消注册过程。
5. 生命周期钩子将实例置于等待状态 (`Terminating:Wait`)，然后执行自定义操作。

   该实例将保持等待状态，直到您完成生命周期操作，或者直至超时期结束（预设情况下为 1 小时）。完成生命周期钩子或超时时段过期后，实例将转换到下一个状态 (`Terminating:Proceed`)。

6. 实例已终止。

Important

暖池中的实例也有自己的生命周期以及相对应的等待状态，如 暖池中实例的生命周期状态转换 (p. 217) 中所述。

做好准备向 Auto Scaling 组添加生命周期钩子

在向 Auto Scaling 组添加生命周期钩子之前，请确保正确用户数据脚本或通知目标。

- 要在实例启动时运行用户数据脚本以对实例执行自定义操作，无需配置通知目标。但是，您必须已创建指定用户数据脚本并将其与 Auto Scaling 组关联的启动模板或启动配置。有关用户数据脚本的更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 启动时对 Linux 实例运行命令。
- 要在生命周期操作完成时向 Amazon EC2 Auto Scaling 发出信号，则必须向该脚本添加 CompleteLifecycleAction API 调用，并且您必须使用允许 Auto Scaling 实例调用此 API 的策略手动创建 IAM 角色。您的启动模板或启动配置必须使用在启动时附加到 Amazon EC2 实例的 IAM 实例配置文件指定此角色。有关更多信息，请参阅 完成生命周期操作 (p. 201) 和 在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)：
- 要使用 Lambda 之类的服务执行自定义操作，您必须首先创建一条 EventBridge 规则并将某个 Lambda 函数指定为该规则的目标。有关更多信息，请参阅为生命周期通知配置通知目标 (p. 194)。
- 要允许 Lambda 在生命周期操作完成时向 Amazon EC2 Auto Scaling 发出信号，您必须向函数代码添加 CompleteLifecycleAction API 调用。您还必须将 IAM policy 附加到函数的执行角色，以授予 Lambda 完成生命周期操作的权限。有关更多信息，请参阅教程：配置调用 Lambda 函数的生命周期钩子 (p. 207)。
- 要使用 Amazon SNS 或 Amazon SQS 之类的服务执行自定义操作，您必须已创建 SNS 主题或 SQS 队列并准备好其 Amazon Resource Name (ARN)。您还必须已创建允许 Amazon EC2 Auto Scaling 访问您的 SNS 主题或 SQS 目标的 IAM 角色，并已准备好其 ARN。有关更多信息，请参阅为生命周期通知配置通知目标 (p. 194)。

  Note

  默认情况下，当您在控制台中添加生命周期钩子时，Amazon EC2 Auto Scaling 会向 Amazon EventBridge 发送生命周期事件通知。建议的最佳实践是使用 EventBridge 或用户数据脚本。要创建直接向 Amazon SNS 或 Amazon SQS 发送通知的生命周期钩子，请使用 Amazon CLI、Amazon CloudFormation 或开发工具包来添加生命周期钩子。

为生命周期通知配置通知目标

您可以向 Auto Scaling 组添加生命周期钩子，以便在实例进入等待状态时执行自定义操作。您可以选择目标服务，以根据您的首选开发方法执行这些操作。

第一种方法使用 Amazon EventBridge 调用执行所需操作的 Lambda 函数。第二种方法是创建发布通知的 Amazon Simple Notification Service (Amazon SNS) 主题。客户端可以订阅 SNS 主题并使用支持的协议接收已发布的消息。最后一种方法涉及到使用 Amazon Simple Queue Service (Amazon SQS)，它是分布式应用程序用于通过轮询模型交换消息的消息收发系统。

作为最佳实践，我们建议您使用 EventBridge。发送到 Amazon SNS 和 Amazon SQS 的通知包含的信息与 Amazon EC2 Auto Scaling 发送到 EventBridge 的通知相同。在 EventBridge 之前，标准做法是向 SNS 或 SQS 发送通知，并将其他服务与 SNS 或 SQS 集成以执行编程操作。如今，EventBridge 为您提供了更多目标服务选择，使您可以使用无服务器架构更轻松地处理事件。

以下过程介绍了如何设置通知目标。

请记住，如果您的启动模板或启动配置中有在实例启动时配置实例的用户数据脚本，则无需接收通知即可对实例执行自定义操作。

目录

- 使用 EventBridge 将通知路由到 Lambda (p. 195)
- 使用 Amazon SNS 接收通知 (p. 196)
- 使用 Amazon SQS 接收通知 (p. 197)
- Amazon SNS 和 Amazon SQS 的通知消息示例 (p. 197)

Important

与生命周期钩子一起使用的 EventBridge 规则、Lambda 函数、Amazon SNS 主题和 Amazon SQS 队列必须始终位于您创建 Auto Scaling 组的同一区域。

使用 EventBridge 将通知路由到 Lambda

您可以将 EventBridge 规则配置为在实例进入等待状态时调用 Lambda 函数。Amazon EC2 Auto Scaling 将向 EventBridge 发送关于正在启动或终止的实例的生命周期事件通知，以及可用于控制生命周期操作的令牌。有关这些事件的示例，请参阅 Amazon EC2 Auto Scaling 事件参考 (p. 288)。

Note

在使用 Amazon Web Services Management Console 创建事件规则时，控制台会自动添加所需的 IAM 权限以便向 EventBridge 授予调用 Lambda 函数所需的权限。如果您使用 Amazon CLI 创建事件规则，则需要明确授予此权限。
要详细了解如何在 EventBridge 控制台中创建事件规则，请参阅《Amazon EventBridge 用户指南》中的创建对事件作出反应的 Amazon EventBridge 规则。
- 或者 -
有关面向控制台用户的入门教程，请参阅教程：配置调用 Lambda 函数的生命周期钩子 (p. 207)。此教程演示了如何创建可侦听启动事件并将其写入 CloudWatch Logs 日志的简单 Lambda 函数。

创建调用 Lambda 函数的 EventBridge 规则

1. 通过使用 Lambda 控制台创建 Lambda 函数，并记下其 Amazon Resource Name (ARN)。例如，`arn:aws:lambda:region:123456789012:function:my-function`。您需要 ARN 来创建 EventBridge 目标。有关更多信息，请参阅 Amazon Lambda 开发人员指南中的 Lambda 入门。
2. 要创建匹配实例创建事件的规则，请使用以下 put-rule 命令。

```
aws events put-rule --name my-rule --event-pattern file://pattern.json --state ENABLED
```

以下示例显示了实例启动生命周期操作的 `pattern.json`。将`##`文本替换为您的自动扩缩组的名称。

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "AutoScalingGroupName": [ "my-asg" ]
   }
}
```

如果该命令成功运行，EventBridge 将使用规则的 ARN 作出响应。记下此 ARN。您需要在第 4 步中输入此信息。

要创建与其他事件匹配的规则，请修改事件模式。有关更多信息，请参阅使用 EventBridge 处理 Auto Scaling 事件 (p. 287)。

3. 要指定要用作规则目标的 Lambda 函数，请使用以下 put-targets 命令。

```
aws events put-targets --rule my-rule --targets
 Id=1,Arn=arn:aws:lambda:region:123456789012:function:my-function
```

在上述命令中，*my-rule* 是您在第 2 步中为规则指定的名称，Arn 参数的值是您在第 1 步中创建的函数的 ARN。

4. 要添加允许规则调用您的 Lambda 函数的权限，请使用以下 Lambda add-permission 命令。此命令信任 EventBridge 服务委托人 (events.amazonaws.com)，并将权限范围划定为指定规则。

```
aws lambda add-permission --function-name my-function --statement-id my-unique-id \
  --action 'lambda:InvokeFunction' --principal events.amazonaws.com --source-arn
 arn:aws:events:region:123456789012:rule/my-rule
```

在上述命令中：

- *my-function* 是要将规则用作目标的 Lambda 函数的名称。
- *my-unique-id* 是您定义的唯一标识符，用来描述 Lambda 函数策略中的语句。
- source-arn 是 EventBridge 规则的 ARN。

如果命令成功运行，则您将收到类似于以下内容的输出：

```
{
  "Statement": "{\"Sid\":\"my-unique-id\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"events.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-west-2:123456789012:function:my-function\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:events:us-west-2:123456789012:rule/my-rule\"}}}"
}
```

Statement 值是已添加到 Lambda 函数策略的语句的 JSON 字符串版本。

5. 在遵循以上说明操作后，继续添加生命周期钩子 (p. 199) 作为下一步。

使用 Amazon SNS 接收通知

您可以使用 Amazon SNS 设置通知目标（SNS 主题），以便在生命周期操作发生时接收通知。然后，Amazon SNS 将通知发送给订阅的收件人。确认订阅前，向主题发布的通知不会发送至收件人。

使用 Amazon SNS 设置通知

1. 通过使用 Amazon SNS 控制台或以下 create-topic 命令创建 Amazon SNS 主题。确保该主题与您使用的 Auto Scaling 组位于同一区域。有关更多信息，请参阅 Amazon Simple Notification Service 开发人员指南中的 Amazon SNS 入门。

```
aws sns create-topic --name my-sns-topic
```

2. 记录主题 Amazon Resource Name (ARN)，例如 arn:aws:sns:*region*:*123456789012*:*my-sns-topic*。您需要它来创建生命周期钩子。
3. 创建 IAM 服务角色，以授予 Amazon EC2 Auto Scaling 访问您的 Amazon SNS 通知目标的权限。

向 Amazon EC2 Auto Scaling 授予访问您的 SNS 主题的权限

a. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
b. 在左侧的导航窗格中，选择角色。
c. 选择 Create role（创建角色）。
d. 对于选择可信实体，选择 Amazon 服务。
e. 对于您的使用案例，在其他 Amazon 服务的使用案例下，选择 EC2 Auto Scaling，然后选择 EC2 Auto Scaling 通知访问权限。
f. 选择 Next（下一步）两次，以前往 Name, review, and create（命名、检查和创建）页面。
g. 对于 Role name（角色名称），输入角色的名称（例如 **my-notification-role**），然后选择 Create role（创建角色）。
h. 在 Roles (角色) 页面中，选择刚刚创建的角色以打开 Summary (摘要) 页面。记下角色的 ARN。例如，`arn:aws:iam::123456789012:role/my-notification-role`。您需要它来创建生命周期钩子。

4. 在遵循以上说明操作后，继续添加生命周期钩子 (Amazon CLI) (p. 200) 作为下一步。

使用 Amazon SQS 接收通知

您可以使用 Amazon SQS 设置通知目标以便在生命周期操作开始时接收消息。然后，队列使用者必须轮询 SQS 队列，以便对这些通知执行操作。

Important

FIFO 队列与生命周期挂钩不兼容。

使用 Amazon SQS 设置通知

1. 使用 Amazon SQS 控制台创建 Amazon SQS 队列。确保队列与您正在使用的 Auto Scaling 组位于同一个区域。有关更多信息，请参阅 Amazon Simple Queue Service 开发人员指南中的 Amazon SQS 入门。
2. 记录队列 ARN，例如 `arn:aws:sqs:us-west-2:123456789012:my-sqs-queue`。您需要它来创建生命周期钩子。
3. 创建 IAM 服务角色，以授予 Amazon EC2 Auto Scaling 访问您的 Amazon SQS 通知目标的权限。

   向 Amazon EC2 Auto Scaling 授予您对 SQS 队列的访问权限

   a. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
   b. 在左侧的导航窗格中，选择角色。
   c. 选择 Create role（创建角色）。
   d. 对于选择可信实体，选择 Amazon 服务。
   e. 对于您的使用案例，在其他 Amazon 服务的使用案例下，选择 EC2 Auto Scaling，然后选择 EC2 Auto Scaling 通知访问权限。
   f. 选择 Next（下一步）两次，以前往 Name, review, and create（命名、检查和创建）页面。
   g. 对于 Role name（角色名称），输入角色的名称（例如 **my-notification-role**），然后选择 Create role（创建角色）。
   h. 在 Roles (角色) 页面中，选择刚刚创建的角色以打开 Summary (摘要) 页面。记下角色的 ARN。例如，`arn:aws:iam::123456789012:role/my-notification-role`。您需要它来创建生命周期钩子。

4. 在遵循以上说明操作后，继续添加生命周期钩子 (Amazon CLI) (p. 200) 作为下一步。

Amazon SNS 和 Amazon SQS 的通知消息示例

当实例处于等待状态时，将向 Amazon SNS 或 Amazon SQS 通知目标发布消息。消息包含以下信息：

- LifecycleActionToken — 生命周期操作令牌。
- AccountId – Amazon Web Services 账户 ID。
- AutoScalingGroupName — Auto Scaling 组的名称。
- LifecycleHookName — 生命周期钩子的名称。
- EC2InstanceId — EC2 实例的 ID。
- LifecycleTransition — 生命周期钩子类型。
- NotificationMetadata — 通知元数据。

以下是通知消息示例。

```
Service: AWS Auto Scaling
Time: 2021-01-19T00:36:26.533Z
RequestId: 18b2ec17-3e9b-4c15-8024-ff2e8ce8786a
LifecycleActionToken: 71514b9d-6a40-4b26-8523-05e7ee35fa40
AccountId: 123456789012
AutoScalingGroupName: my-asg
LifecycleHookName: my-hook
EC2InstanceId: i-0598c7d356eba48d7
LifecycleTransition: autoscaling:EC2_INSTANCE_LAUNCHING
NotificationMetadata: hook message metadata
```

测试通知消息示例

首次添加生命周期钩子时，将向通知目标发布测试通知消息。以下是测试通知消息示例。

```
Service: AWS Auto Scaling
Time: 2021-01-19T00:35:52.359Z
RequestId: 18b2ec17-3e9b-4c15-8024-ff2e8ce8786a
Event: autoscaling:TEST_NOTIFICATION
AccountId: 123456789012
AutoScalingGroupName: my-asg
AutoScalingGroupARN: arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:042cba90-
ad2f-431c-9b4d-6d9055bcc9fb:autoScalingGroupName/my-asg
```

> **Note**
>
> 有关从 Amazon EC2 Auto Scaling 传输到 EventBridge 的事件示例，请参阅 Amazon EC2 Auto Scaling 事件参考 (p. 288)。

通过实例元数据检索目标生命周期状态

> **Important**
>
> Amazon EC2 Auto Scaling 于 2022 年 3 月 10 日开始生成目标生命周期状态。如果您的实例在该日期之后转换为其中一个目标生命周期状态，则您的实例元数据中将显示该目标生命周期状态项目。否则，它不存在，并且您会收到 HTTP 404 错误。

您启动的每个 Auto Scaling 实例都经历多个生命周期状态。如果您想对实例上的自定义操作进行微调，以便对特定的生命周期状态转换采取行动，则可以通过实例元数据检索目标生命周期状态来实现此目标。例如，您可以在为暖池中的实例创建生命周期操作时检索目标生命周期状态，以便在重启时将休眠或已停止的实例附加到集群。

Auto Scaling 实例生命周期有两个主要的稳定状态：InService 和 Terminated，还有两个附加的稳定状态：Detached 和 Standby。如果使用暖池，生命周期还有四个额外的稳定状态：Warmed:Hibernated、Warmed:Running、Warmed:Stopped 和 Warmed:Terminated。

当实例准备转换到之前的稳定状态之一时，Amazon EC2 Auto Scaling 会更新实例元数据项目 `autoscaling/target-lifecycle-state` 的值。要从实例内获取目标生命周期状态，必须使用实例元数据服务从实例元数据中检索它。

> **Note**
>
> 实例元数据是有关 Amazon EC2 实例的数据，应用程序可以用它来查询实例信息。实例元数据服务是实例上的组件，本地代码用它来访问实例元数据。本地代码可以包括实例上运行的用户数据脚本或应用程序。

本地代码可以使用以下两种方法之一从正在运行的实例访问实例元数据：实例元数据服务版本 1 (IMDSv1) 或实例元数据服务版本 2 (IMDSv2)。IMDSv2 使用面向会话的请求，并防范了多种类型的漏洞，这些漏洞可用于尝试访问实例元数据。有关这两种方法的详细信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的使用 IMDSv2。

IMDSv2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/autoscaling/target-lifecycle-state
```

IMDSv1

```
curl http://169.254.169.254/latest/meta-data/autoscaling/target-lifecycle-state
```

下面是示例输出。

```
InService
```

目标生命周期状态是实例转换到的状态。当前生命周期状态是实例所处的状态。在生命周期操作完成并且实例完成向目标生命周期状态的转换之后，这两个状态可能是相同的。您无法从实例元数据中检索实例的当前生命周期状态。

有关向您展示如何在使用目标生命周期状态的用户数据脚本中使用自定义操作创建生命周期钩子的教程，请参阅 教程：配置用户数据以通过实例元数据检索目标生命周期状态 (p. 202)。

有关可以用来配置或管理实例的实例元数据类别的更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的实例元数据类别。有关检索实例元数据的更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的检索实例元数据。

添加生命周期钩子

要将 Auto Scaling 实例置于等待状态并对它们执行自定义操作，您可以向 Auto Scaling 组添加生命周期钩子。自定义操作将在实例启动时或其终止之前执行。实例将保持等待状态，直到您完成生命周期操作，或者直到超时时段结束。

从 Amazon Web Services Management Console 创建 Auto Scaling 组后，您可以向其添加一个或多个生命周期钩子，最多可以添加 50 个生命周期钩子。您也可以使用 Amazon CLI、Amazon CloudFormation 或开发工具包在创建 Auto Scaling 组时向其添加生命周期钩子。

默认情况下，当您在控制台中添加生命周期钩子时，Amazon EC2 Auto Scaling 会向 Amazon EventBridge 发送生命周期事件通知。建议的最佳实践是使用 EventBridge 或用户数据脚本。要创建直接向 Amazon SNS 或 Amazon SQS 发送通知的生命周期钩子，您可以使用 put-lifecycle-hook 命令，如此主题中的示例所示。

目录

- 添加生命周期钩子（控制台）(p. 200)
- 添加生命周期钩子 (Amazon CLI) (p. 200)

添加生命周期钩子（控制台）

请按照以下步骤向您的 Auto Scaling 组添加生命周期钩子。要创建用于横向扩展（实例启动）和横向缩减（实例终止）的生命周期钩子，您必须创建两个单独的钩子。

如 做好准备向 Auto Scaling 组添加生命周期钩子 (p. 194) 中所述，请在开始之前确认已设置自定义操作。

添加生命周期挂钩

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling 组页面底部打开一个拆分窗格。
3. 在 Instance management (实例管理) 选项卡的 Lifecycle hooks (生命周期挂钩) 中，选择 Create lifecycle hook (创建生命周期挂钩)。
4. 要定义生命周期挂钩，请执行以下操作：

   a. 对于 Lifecycle hook name (生命周期挂钩名称)，请指定生命周期挂钩的名称。

   b. 对于 Lifecycle transition (生命周期转换)，请选择 Instance launch (实例启动) 或 Instance terminate (实例终止)。

   c. 对于检测信号超时时间，请在钩子超时之前，指定实例在横向扩展或横向缩减时保持等待状态的时长。范围从 30 到 7200 秒。

      Note

      默认超时时间为 3600 秒（一小时）。设置的超时时段越长，完成自定义操作的时间越长。如果您在超时时段结束之前完成操作，可以发送 complete-lifecycle-action 命令，以允许实例继续到下一状态。

   d. 对于默认结果，指定在生命周期钩子超时结束或发生意外故障时要执行的操作。您可以选择放弃（原定设置）或继续。

      - 如果实例正在启动，继续指示您的操作已成功，并且 Amazon EC2 Auto Scaling组可将实例投入使用。否则，放弃指示您的自定义操作未成功，并且 Amazon EC2 Auto Scaling 可终止实例。
      - 如果实例正在终止，放弃和继续都允许终止实例。不过，放弃将停止任何剩余操作（例如，其他生命周期钩子），而继续将允许完成任何其他生命周期钩子。

   e. （可选）对于通知元数据，请指定您希望在 Amazon EC2 Auto Scaling 向通知目标发送消息时包含的其他信息。
5. 选择 Create（创建）。

添加生命周期钩子 (Amazon CLI)

使用 put-lifecycle-hook 命令创建和更新生命周期钩子。

要执行扩展操作，请使用以下命令。

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-launch-hook  \
  --auto-scaling-group-name my-asg \
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING
```

要执行缩减操作，请使用以下命令。

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook  \
  --auto-scaling-group-name my-asg \
  --lifecycle-transition autoscaling:EC2_INSTANCE_TERMINATING
```

要使用 Amazon SNS 或 Amazon SQS 接收通知，请添加 `--notification-target-arn` 和 `--role-arn` 选项。

以下示例中创建了一个生命周期钩子，用于指定名为的 *my-sns-topic* 的 SNS 主题作为通知目标。

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook  \
  --auto-scaling-group-name my-asg \
  --lifecycle-transition autoscaling:EC2_INSTANCE_TERMINATING \
  --notification-target-arn arn:aws:sns:region:123456789012:my-sns-topic \
  --role-arn arn:aws:iam::123456789012:role/my-notification-role
```

该主题将使用以下键/值对接收测试通知。

```
"Event": "autoscaling:TEST_NOTIFICATION"
```

默认情况下，put-lifecycle-hook 命令会创建一个生命周期钩子，检测信号超时时间为 3600 秒（一小时）。

要更改现有生命周期钩子的检测信号超时时间，请添加 `--heartbeat-timeout` 选项，如以下示例所示。

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook \
  --auto-scaling-group-name my-asg --heartbeat-timeout 120
```

如果实例已处于等待状态，则可以通过记录检测信号、使用 record-lifecycle-action-heartbeat CLI 命令来防止生命周期钩子超时。这会将超时时间增加到您创建生命周期挂钩时指定的超时值。如果您在超时时段结束之前完成操作，可以发送 complete-lifecycle-action CLI 命令，以允许实例继续到下一状态。有关更多信息以及示例，请参阅 完成生命周期操作 (p. 201)。

完成生命周期操作

当 Auto Scaling 组响应生命周期事件时，它会将实例置于等待状态并发送事件通知。当实例处于等待状态时，您可以执行自定义操作。

目录

- 完成生命周期操作（手动）(p. 201)
- 完成生命周期操作（自动）(p. 202)

完成生命周期操作（手动）

以下过程适用于命令行界面，在控制台中不受支持。必须替换的信息（如实例 ID 或 Auto Scaling 组的名称）以斜体显示。

完成生命周期操作 (Amazon CLI)

1. 如果需要更多时间来完成自定义操作，请使用 record-lifecycle-action-heartbeat 命令重新启动超时时段并使实例保持等待状态。例如，如果超时时段为一小时，而您在 30 分钟后调用该命令，则实例将继续保持等待状态一小时（总共为 90 分钟）。

   您可以指定随通知 (p. 197)一起接收的生命周期操作令牌，如以下命令所示。

```
aws autoscaling record-lifecycle-action-heartbeat --lifecycle-hook-name my-launch-hook
 \
  --auto-scaling-group-name my-asg --lifecycle-action-
token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

或者，您可以指定随通知 (p. 197)一起接收的实例的 ID，如以下命令所示。

```
aws autoscaling record-lifecycle-action-heartbeat --lifecycle-hook-name my-launch-hook
 \
  --auto-scaling-group-name my-asg --instance-id i-1a2b3c4d
```

2. 如果您在超时时段结束前已完成自定义操作，请使用 complete-lifecycle-action 命令，以便 Auto Scaling 组能够继续启动或终止实例。您可以指定生命周期操作令牌，如以下命令所示。

```
aws autoscaling complete-lifecycle-action --lifecycle-action-result CONTINUE \
  --lifecycle-hook-name my-launch-hook --auto-scaling-group-name my-asg \
  --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

或者，您可以指定实例的 ID，如以下命令所示。

```
aws autoscaling complete-lifecycle-action --lifecycle-action-result CONTINUE \
  --instance-id i-1a2b3c4d --lifecycle-hook-name my-launch-hook \
  --auto-scaling-group-name my-asg
```

完成生命周期操作（自动）

如果您拥有在实例启动后配置这些实例的用户数据脚本，则无需手动完成生命周期操作。可以将 complete-lifecycle-action 命令添加到脚本中。该脚本可以从实例元数据中检索实例 ID，并在引导启动脚本成功完成后向 Amazon EC2 Auto Scaling 发出信号。

如果您尚未这样做，请更新脚本，从实例元数据中检索实例的实例 ID。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的检索实例元数据。

如果您使用 Lambda，则还可以在函数的代码中设置回调，以便在自定义操作成功时能让实例的生命周期继续。有关更多信息，请参阅教程：配置调用 Lambda 函数的生命周期钩子 (p. 207)。

教程：配置用户数据以通过实例元数据检索目标生命周期状态

为生命周期钩子创建自定义操作的一种常见方法是使用 Amazon EC2 Auto Scaling 发送给其他服务（例如 Amazon EventBridge）的通知。但是，您可以避免创建额外的基础设施，方法是使用用户数据脚本将配置实例并完成生命周期操作的代码移动到实例本身中。

以下教程介绍如何开始使用用户数据脚本和实例元数据。您可以使用读取您组中实例的目标周期状态 (p. 198)并在实例生命周期的特定阶段执行回调操作以继续启动过程的用户数据脚本来创建基本 Auto Scaling 组配置。

目录

- 步骤 1：创建具有完成生命周期操作权限的 IAM 角色 (p. 203)
- 步骤 2：创建启动模板并包含 IAM 角色和用户数据脚本 (p. 203)
- 步骤 3：创建 Auto Scaling 组 (p. 205)
- 步骤 4：添加生命周期钩子 (p. 205)

- 步骤 5：测试和验证功能 (p. 206)
- 步骤 6：清除 (p. 206)

步骤 1：创建具有完成生命周期操作权限的 IAM 角色

使用 Amazon CLI 或 Amazon 开发工具包发送回调以完成生命周期操作时，您必须使用具有完成生命周期操作权限的 IAM 角色。

创建策略

1. 打开 IAM 控制台的策略页面，然后选择创建策略。
2. 请选择 JSON 选项卡。
3. 在策略文档框中，将以下策略文档复制并粘贴到框中。将*####*替换为您的账号和您要创建的 Auto Scaling 组的名称 (**`TestAutoScalingEvent-group`**)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction"
      ],
      "Resource":
 "arn:aws:autoscaling:*:123456789012:autoScalingGroup:*:autoScalingGroupName/
TestAutoScalingEvent-group"
    }
  ]
}
```

4. 选择 Next: Tags，然后选择 Next:Review。
5. 对于 Name (名称)，请输入 **`TestAutoScalingEvent-policy`**。选择 Create policy（创建策略）。

完成创建策略之后，您可以创建一个使用该策略的角色。

创建角色

1. 在左侧的导航窗格中，选择角色。
2. 选择 Create role（创建角色）。
3. 对于选择可信实体，选择 Amazon 服务。
4. 对于您的使用案例，选择 EC2，然后选择 Next（下一步）。
5. 在 Attach permissions（附加权限）下，选择您创建的策略（TestAutoScalingEvent-policy）。然后选择下一步。
6. 在 Name, review, and create（命名、检查并创建）页面上，对于 Role name（角色名称），输入 **`TestAutoScalingEvent-role`**，然后选择 Create role（创建角色）。

步骤 2：创建启动模板并包含 IAM 角色和用户数据脚本

创建用于 Auto Scaling 组的启动模板。包含您创建的 IAM 角色和提供的示例用户数据脚本。

创建启动模板

1. 打开 Amazon EC2 控制台上的启动模板页面。

2. 选择Create launch template（创建启动模板）。
3. 对于 Launch template name（启动模板名称），输入 **TestAutoScalingEvent-template**。
4. 在 Auto Scaling guidance（Auto Scaling 指导）下，选中复选框。
5. 对于 Application and OS Images (Amazon Machine Image) [应用程序和操作系统镜像（Amazon Machine Image）]，请从 Quick Start（快速启动）列表中选择 Amazon Linux 2 (HVM)、SSD Volume Type（SSD 卷类型）、64-bit (x86) [64 位（x86）]。
6. 对于 Instance type（实例类型），选择一种 Amazon EC2 实例类型（例如“t2.micro”）。
7. 对于高级详细信息，展开该节以查看字段。
8. 对于 IAM instance profile（IAM 实例配置文件），选择您 IAM 角色的 IAM 实例配置文件名称（TestAutoScalingEvent-role）。实例配置文件是 IAM 角色的容器，在实例启动时允许 Amazon EC2 将 IAM 角色传递到实例。

   使用 IAM 控制台创建 IAM 角色时，控制台自动创建实例配置文件，按相应的角色为文件命名。
9. 对于 User data（用户数据），将下面的实例用户数据脚本粘贴到字段。将 `group_name` 的示例文本替换为您要创建的自动扩缩组的名称，将 `region` 替换为您希望您的自动扩缩组使用的 Amazon Web Services 区域 区域。

```
#!/bin/bash

function get_target_state {
    echo $(curl -s http://169.254.169.254/latest/meta-data/autoscaling/target-
lifecycle-state)
}

function get_instance_id {
    echo $(curl -s http://169.254.169.254/latest/meta-data/instance-id)
}

function complete_lifecycle_action {
    instance_id=$(get_instance_id)
    group_name='TestAutoScalingEvent-group'
    region='us-west-2'

    echo $instance_id
    echo $region
    echo $(aws autoscaling complete-lifecycle-action \
      --lifecycle-hook-name TestAutoScalingEvent-hook \
      --auto-scaling-group-name $group_name \
      --lifecycle-action-result CONTINUE \
      --instance-id $instance_id \
      --region $region)
}

function main {
    while true
    do
        target_state=$(get_target_state)
        if [ \"$target_state\" = \"InService\" ]; then
            # Change hostname
            export new_hostname="${group_name}-$instance_id"
            hostname $new_hostname
            # Send callback
            complete_lifecycle_action
            break
        fi
        echo $target_state
        sleep 5
    done
}
```

```
main
```

此简单用户数据脚本将执行以下操作：

- 调用实例元数据以从实例元数据中检索目标生命周期状态和实例 ID
- 重复检索目标生命周期状态，直到它变为 `InService`
- 如果目标生命周期状态为 `InService`，则将实例的主机名更改为实例 ID 前面加上 Auto Scaling 组名称
- 通过调用 complete-lifecycle-action CLI 命令向 Amazon EC2 Auto Scaling 发出信号以 `CONTINUE` EC2 启动过程来发送回调

10. 选择Create launch template（创建启动模板）。
11. 在确认页面上，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

Note

有关可用作开发用户数据脚本的其他参考示例，请参阅 Amazon EC2 Auto Scaling 的 GitHub 存储库。

步骤 3：创建 Auto Scaling 组

创建启动模板后，创建 Auto Scaling 组。

创建 自动扩缩组

1. 在 Choose launch template or configuration（选择启动模板或配置）页面上，对于 Auto Scaling group name（Auto Scaling 组名称），输入 Auto Scaling 组的名称 (**`TestAutoScalingEvent-group`**)。
2. 选择 Next（下一步）转至 Choose instance launch options（选择实例启动选项）页面。
3. 对于 Network（网络），选择 VPC。
4. 对于 Availability Zones and subnets（可用区和子网），请从一个或多个可用区中选择一个或多个子网。
5. 在 Instance type requirements（实例类型要求）部分中，使用默认设置简化此步骤。（请勿覆盖启动模板。）在本教程中，您将仅使用启动模板中指定的实例类型启动一个按需型实例。
6. 在屏幕的底部选择 Skip to review（跳至审核）。
7. 在 Review（审核）页面中，检查 Auto Scaling 组的详细信息，然后选择 Create Auto Scaling group（创建 Auto Scaling 组）。

步骤 4：添加生命周期钩子

添加生命周期钩子以将实例保持在等待状态，直到您的生命周期操作完成。

添加生命周期挂钩

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling 组页面底部打开一个拆分窗格。
3. 在下方窗格中，在实例管理选项卡的生命周期钩子中，选择创建生命周期钩子。
4. 要定义生命周期挂钩，请执行以下操作：

   a. 对于生命周期钩子名称，请输入 **`TestAutoScalingEvent-hook`**。

b. 对于生命周期转换，请选择实例启动。

c. 对于 Heartbeat timeout（检测信号超时），输入 **300** 以获取等待用户数据脚本回调的秒数。

d. 对于默认结果，请选择放弃。如果钩子超时而未收到来自您用户数据脚本的回调，Auto Scaling 组将终止新实例。

e. （可选）将 Notification metadata（通知元数据）留空。

5. 选择 Create（创建）。

步骤 5：测试和验证功能

要测试功能，请通过将 Auto Scaling 组的所需容量增加 1 来更新 Auto Scaling 组。用户数据脚本运行并在实例启动后立即开始检查实例的目标生命周期状态。当目标生命周期状态为 `InService` 时，脚本会更改主机名并发送回调操作。完成此操作通常仅需要几秒钟时间。

增加您的 Auto Scaling 组的大小

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中您的 Auto Scaling 组旁边的复选框。在下方窗格中查看详细信息，同时仍然可以看到上方窗格的顶部几行。
3. 在下方窗格中的详细信息选项卡上，选择组详细信息、编辑。
4. 对于 Desired capacity (所需容量)，将当前值增加 1。
5. 选择 Update（更新）。启动实例时，上方窗格中的状态列会显示正在更新容量状态。

在增加所需容量后，您可以通过扩缩活动的描述来验证您的实例是否已成功启动且未终止。

查看扩展活动

1. 返回到 Auto Scaling 组页面并选择您的组。
2. 在活动选项卡上的活动历史记录下，状态列显示您的 Auto Scaling 组是否已成功启动实例。
3. 如果用户数据脚本失败，则在超时时段过后，您会看到状态为 `Canceled` 且状态消息为 `Instance failed to complete user's Lifecycle Action: Lifecycle Action with token e85eb647-4fe0-4909-b341-a6c42EXAMPLE was abandoned: Lifecycle Action Completed with ABANDON Result` 的扩缩活动。

步骤 6：清除

如果您已完成为本教程创建的资源，请使用以下步骤将其删除。

删除生命周期钩子

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中您的 Auto Scaling 组旁边的复选框。
3. 在实例管理选项卡的生命周期钩子中，选择生命周期钩子 (`TestAutoScalingEvent-hook`)。
4. 依次选择 Actions（操作）和 Delete（删除）。
5. 再次选择删除以确认。

要删除启动模板

1. 打开 Amazon EC2 控制台上的 Launch templates page（启动模板页面）。

2. 选择启动模板 (`TestAutoScalingEvent-template`)，然后依次选择 Actions（操作）、Delete template（删除模板）。
3. 当系统提示进行确认时，键入 **`Delete`** 以确认删除指定启动模板，然后选择 Delete（删除）。

如果您已完成使用此示例 Auto Scaling 组，请将其删除。您还可以删除您创建的 IAM 角色和权限策略。

要删除 Auto Scaling 组

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中 Auto Scaling 组 (`TestAutoScalingEvent-group`) 旁边的复选框并选择 Delete（删除）。
3. 当系统提示进行确认时，键入 **`delete`** 以确认删除指定自动扩缩组，然后选择 Delete（删除）。

   Name（名称）列中的加载图标指示 Auto Scaling 组正在被删除。终止实例并删除组需要几分钟时间。

要删除 IAM 角色

1. 打开 IAM 控制台的 Roles page（角色页面）。
2. 选择函数的角色 (`TestAutoScalingEvent-role`)。
3. 选择 Delete (删除)。
4. 在系统提示进行确认时，键入角色的名称，然后选择 Delete（删除）。

删除 IAM policy

1. 打开 IAM 控制台的 Policies（策略）页面。
2. 选择您创建的策略 (`TestAutoScalingEvent-policy`)。
3. 依次选择 Actions（操作） 和 Delete（删除）。
4. 在系统提示进行确认时，键入策略的名称，然后选择 Delete（删除）。

教程：配置调用 Lambda 函数的生命周期钩子

在此练习中，您将创建一个包含筛选条件模式的 Amazon EventBridge 规则，该规则将在匹配时调用一个作为规则目标的 Amazon Lambda 函数。我们提供了要使用的筛选条件模式和示例函数代码。

如果一切配置正确，则在本教程结束时，Lambda 函数将在实例启动时执行自定义操作。自定义操作只是在与 Lambda 函数关联的 CloudWatch Logs 日志流中记录事件。

Lambda 函数还执行回调，以便在此操作成功时继续执行实例的生命周期，但允许实例放弃启动并在操作失败时终止。

有关使用 EventBridge 的更多信息，请参阅使用 EventBridge 处理 Auto Scaling 事件 (p. 287)。

目录

- 先决条件 (p. 208)
- 步骤 1：创建具有完成生命周期操作权限的 IAM 角色 (p. 208)
- 步骤 2：创建 Lambda 函数 (p. 208)
- 步骤 3：创建 EventBridge 规则 (p. 209)
- 步骤 4：添加生命周期钩子 (p. 210)
- 步骤 5：测试和验证事件 (p. 211)
- 步骤 6：后续步骤 (p. 211)

- 步骤 7：清除 (p. 211)

先决条件

在开始本教程前，请创建 Auto Scaling 组（如果您还没有 Auto Scaling 组）。要创建自动扩缩组，请打开 Amazon EC2 控制台中的自动扩缩组页面，然后选择 Create Auto Scaling group（创建自动扩缩组）。

步骤 1：创建具有完成生命周期操作权限的 IAM 角色

在创建 Lambda 函数之前，必须首先创建执行角色和权限策略，以允许 Lambda 完成生命周期钩子。

创建策略

1. 打开 IAM 控制台的策略页面，然后选择创建策略。
2. 请选择 JSON 选项卡。
3. 在策略文档框中，将以下策略文档粘贴到框中，并将*##*文本替换为 Auto Scaling 组的账号和名称。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction"
      ],
      "Resource":
 "arn:aws:autoscaling:*:123456789012:autoScalingGroup:*:autoScalingGroupName/my-asg"
    }
  ]
}
```

4. 选择 Next: Tags，然后选择 Next:Review。
5. 对于 Name (名称)，请输入 **LogAutoScalingEvent-policy**。选择 Create policy（创建策略）。

完成创建策略之后，您可以创建一个使用该策略的角色。

创建角色

1. 在左侧的导航窗格中，选择角色。
2. 选择 Create role（创建角色）。
3. 对于选择可信实体，选择 Amazon 服务。
4. 对于您的使用案例，选择 Lambda，然后选择 Next（下一步）。
5. 在 Add permissions（添加权限）下，选择您创建的策略（LogAutoScalingEvent-policy）和名为 AWSLambdaBasicExecutionRole 的策略。然后选择下一步。

   Note

   AWSLambdaBasicExecutionRole 策略具有函数将日志写入 CloudWatch Logs 所需的权限。
6. 在 Name, review, and create（命名、检查并创建）页面上，对于 Role name（角色名称），输入 **LogAutoScalingEvent-role**，然后选择 Create role（创建角色）。

步骤 2：创建 Lambda 函数

创建 Lambda 函数以用作事件的目标。当 Amazon EC2 Auto Scaling 发送匹配事件时，EventBridge 会调用在 Node.js 中写入的 Lambda 函数。

创建 Lambda 函数

1. 打开 Lambda 控制台的 Functions（函数）页面。
2. 选择 Create function（创建函数），然后选择 Author from scratch（从头开始创作）。
3. 在基本信息 下，对于函数名称，输入 **LogAutoScalingEvent**。
4. 选择更改默认执行角色，然后对于执行角色，选择使用现有角色。
5. 对于现有角色，选择 LogAutoScalingEvent-role。
6. 保留其他默认值。
7. 选择 Create function（创建函数）。您将返回到函数的代码和配置。
8. 当 LogAutoScalingEvent 函数在控制台中保持打开状态时，在编辑器中的 Function code（函数代码）下，将以下示例代码复制到名为 index.js 的文件中。

```
var aws = require("aws-sdk");
exports.handler = (event, context, callback) => {
    console.log('LogAutoScalingEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    var autoscaling = new aws.AutoScaling({region: event.region});
    var eventDetail = event.detail;
    var params = {
        AutoScalingGroupName: eventDetail['AutoScalingGroupName'], /* required */
        LifecycleActionResult: 'CONTINUE', /* required */
        LifecycleHookName: eventDetail['LifecycleHookName'], /* required */
        InstanceId: eventDetail['EC2InstanceId'],
        LifecycleActionToken: eventDetail['LifecycleActionToken']
    };
    var response;
    autoscaling.completeLifecycleAction(params, function(err, data) {
        if (err) {
            console.log(err, err.stack); // an error occurred
            response = {
                statusCode: 500,
                body: JSON.stringify('ERROR'),
            };
        } else {
            console.log(data);           // successful response
            response = {
                statusCode: 200,
                body: JSON.stringify('SUCCESS'),
            };
        }
    });
    return response;
};
```

此代码只记录事件，以便在本教程结束时，您可以在与此 Lambda 函数关联的 CloudWatch Logs 日志流中看到事件。

9. 选择 Deploy（部署）。

步骤 3：创建 EventBridge 规则

创建 EventBridge 规则以运行您的 Lambda 函数。

使用控制台创建规则

1. 打开 EventBridge 控制台。
2. 在导航窗格中，选择 Rules (规则)。
3. 选择 Create rule (创建规则)。

4. 对于定义规则详细信息，请执行以下操作：
   a. 对于 Name（名称），请输入 **LogAutoScalingEvent-rule**。
   b. 对于 Event bus（事件总线），选择 default（默认）。当您账户中的某个 Amazon Web Service生成一个事件时，它始终会发送到您账户的默认事件总线。
   c. 对于 Rule type（规则类型），选择 Rule with an event pattern（具有事件模式的规则）。
   d. 选择 Next（下一步）。
5. 对于 Build event pattern（构建事件模式），执行以下操作：
   a. 对于 Event source（事件源），选择 Amazon 事件或 EventBridge 合作伙伴事件。
   b. 对于 Event pattern（事件模式），执行以下操作：
      i. 对于事件源，选择 Amazon Web Services。
      ii. 对于 Amazon Web Service，选择 Auto Scaling。
      iii. 对于 Event Type (事件类型)，选择 Instance Launch and Terminate (实例启动和终止)。
      iv. 默认情况下，该规则会与任何横向缩减或横向扩展事件匹配。要创建在发生横向扩展事件并且因生命周期挂钩而使实例被置于等待状态时通知您的规则，请选择 Specific instance event(s)（特定实例事件），然后选择 EC2 Instance-launch Lifecycle Action（EC2 实例启动生命周期操作）。
      v. 默认情况下，该规则与区域中任何 Auto Scaling 组匹配。若要使该规则与特定 Auto Scaling 组匹配，请选择 特定组名称并选择一个或多个 Auto Scaling 组。
      vi. 选择 Next（下一步）。
6. 对于 Select target(s)（选择目标），请执行以下操作：
   a. 对于 Target types（目标类型），选择 Amazon Web Service。
   b. 对于 Select a target（选择目标），选择 Lambda function（Lambda 函数）。
   c. 对于 Function（函数），选择 LogAutoScalingEvent。
   d. 选择 Next（下一步）两次。
7. 请在 Review and create（审核和创建）页面上，选择 Create（创建）。

步骤 4：添加生命周期钩子

在本节中，您将添加一个生命周期钩子，以便 Lambda 在启动时在实例上运行函数。

添加生命周期挂钩

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling 组页面底部打开一个拆分窗格。
3. 在下方窗格中，在实例管理选项卡的生命周期钩子中，选择创建生命周期钩子。
4. 要定义生命周期挂钩，请执行以下操作：
   a. 对于生命周期钩子名称，请输入 **LogAutoScalingEvent-hook**。
   b. 对于生命周期转换，请选择实例启动。
   c. 对于检测信号超时，输入 **300** 以获取等待 Lambda 函数回调的秒数。
   d. 对于默认结果，请选择放弃。这意味着，如果钩子超时而未收到 Lambda 函数的回调，Auto Scaling 组将终止新实例。
   e. （可选）将通知元数据留空。我们传递给 EventBridge 的事件数据包含调用 Lambda 函数所需的所有信息。
5. 选择 Create（创建）。

步骤 5：测试和验证事件

要测试事件，请通过将 Auto Scaling 组的所需容量增加 1 来更新 Auto Scaling 组。您的 Lambda 函数在增加所需容量后几秒钟内调用。

增加您的 Auto Scaling 组的大小

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中 Auto Scaling 组旁边的复选框可在下方窗格中查看详细信息，并且仍然可以看到上方窗格的顶行。
3. 在下方窗格中的详细信息选项卡上，选择组详细信息、编辑。
4. 对于 Desired capacity (所需容量)，将当前值增加 1。
5. 选择 Update（更新）。启动实例时，上方窗格中的状态列会显示正在更新容量状态。

增加所需的容量后，可验证是否已调用 Lambda 函数。

查看 Lambda 函数的输出

1. 打开 CloudWatch 控制台的 Log groups page（日志组页面）。
2. 选择您的 Lambda 函数 (`/aws/lambda/LogAutoScalingEvent`) 的日志组的名称。
3. 选择日志流的名称，以查看由生命周期操作的函数提供的数据。

接下来，您可以通过扩展活动的描述来验证您的实例是否已成功启动。

查看扩展活动

1. 返回到 Auto Scaling 组页面并选择您的组。
2. 在活动选项卡上的活动历史记录下，状态列显示您的 Auto Scaling 组是否已成功启动实例。
   - 如果操作成功，扩展活动的状态将为“成功”。
   - 如果失败，在等待几分钟后，您将看到状态为“已取消”的扩展活动，并显示状态消息“实例无法完成用户的生命周期操作：使用令牌 e85eb647-4fe0-4909-b341-a6c42EXAMPLE 的生命周期操作已放弃：使用“放弃”结果完成的生命周期操作”。

减少您的 Auto Scaling 组的大小

如果您不需要您为此测试启动的其他实例，则可以打开详细信息选项卡并将所需容量减少 1。

步骤 6：后续步骤

现在您已完成本教程，您可以尝试创建终止生命周期钩子。如果 Auto Scaling 组中的实例终止，则向 EventBridge 发送事件。有关实例终止时发送的事件的信息，请参阅 EC2 实例终止生命周期操作 (p. 290)。

步骤 7：清除

如果您已完成仅为本教程创建的资源，请使用以下步骤将其删除。

删除生命周期钩子

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中您的 Auto Scaling 组旁边的复选框。
3. 在实例管理选项卡的生命周期钩子中，选择生命周期钩子 (`LogAutoScalingEvent-hook`)。

4. 依次选择 Actions（操作）和 Delete（删除）。
5. 再次选择删除以确认。

删除 Amazon EventBridge 规则

1. 打开 Amazon EventBridge 控制台中的规则页面。
2. 在事件总线下，选择与规则 (`Default`) 关联的事件总线。
3. 然后选中您的规则 (`LogAutoScalingEvent-rule`) 旁边的复选框。
4. 选择 Delete (删除)。
5. 在系统提示进行确认时，键入规则的名称，然后选择 Delete（删除）。

如果您已完成使用此示例函数，请将其删除。您还可以删除存储函数日志的日志组以及您创建的执行角色和权限策略。

要删除 Lambda 函数

1. 打开 Lambda 控制台的 Functions（函数）页面。
2. 选择函数 (`LogAutoScalingEvent`)。
3. 依次选择 Actions（操作）和 Delete（删除）。
4. 当系统提示进行确认时，键入 **`delete`** 以确认删除指定函数，然后选择 Delete（删除）。

删除日志组

1. 打开 CloudWatch 控制台的 Log groups page（日志组页面）。
2. 选择函数的日志组 (`/aws/lambda/LogAutoScalingEvent`)。
3. 依次选择 Actions（操作）和 Delete log group(s)（删除日志组）。
4. 在 Delete log group(s)（删除日志组）对话框中，选择 Delete（删除）。

删除执行角色

1. 打开 IAM 控制台的 Roles page（角色页面）。
2. 选择函数的角色 (`LogAutoScalingEvent-role`)。
3. 选择 Delete (删除)。
4. 在系统提示进行确认时，键入角色的名称，然后选择 Delete（删除）。

删除 IAM policy

1. 打开 IAM 控制台的 Policies（策略）页面。
2. 选择您创建的策略 (`LogAutoScalingEvent-policy`)。
3. 依次选择 Actions（操作）和 Delete（删除）。
4. 在系统提示进行确认时，键入策略的名称，然后选择 Delete（删除）。

Amazon EC2 Auto Scaling 的暖池

暖池使您能够减少引导时间非常长的应用程序的延迟，例如，因为实例需要将大量数据写入磁盘。使用暖池，您不再需要过度配置 Auto Scaling 组来管理延迟以提高应用程序性能。有关更多信息，请参阅以下博客文章 利用 EC2 Auto Scaling 暖池更快地扩展您的应用程序。

Important

在不需要的情况下创建暖池可能会导致不必要的成本。如果您的首次启动时间不会给您的应用程序带来明显的延迟问题，则您可能不需要使用暖池。

目录

- 核心概念 (p. 213)
- 先决条件 (p. 214)
- 创建暖池 (p. 215)
- 更新暖池 (p. 215)
- 删除暖池 (p. 215)
- 限制 (p. 216)
- 将生命周期钩子与暖池一起使用 (p. 216)
- 查看运行状况检查状态以及运行状况检查失败的原因 (p. 218)
- 使用 Amazon CLI 创建和管理暖池的示例 (p. 220)

核心概念

在您开始之前，请熟悉以下核心概念：

暖池

暖池是预初始化的 EC2 实例的池，它位于 Auto Scaling 组旁边。无论何时您的应用程序需要向外扩展，自动扩缩组都可以在暖池上绘制，以满足其新的所需容量。它可帮助您确保实例准备好快速开启为应用程序提供流量，从而加快对横向扩展事件的响应。当实例离开暖池时，它们将计入该组的所需容量。这称为热启动。

当实例处于预热池中时，仅当处于 `InService` 状态的实例的指标值大于扩缩策略的告警阈值上限（与目标跟踪扩缩策略的目标利用率相同）时，扩缩策略才会横向扩展。

暖池大小

默认情况下，暖池的大小以 Auto Scaling 组的最大容量与其所需容量之差来计算。例如，如果 Auto Scaling 组的所需容量为 6 且最大容量为 10，则当您首次设置暖池并且该池正在初始化时，暖池的大小将为 4。

要单独指定暖池的最大容量，请设置一个大于该组当前容量的最大预热容量值。如果您设置了最大预热容量值，则暖池的大小将以该组的最大预热容量与当前所需容量之差来计算。例如，如果您的 Auto Scaling 组的所需容量为 6，最大容量为 10，最大预热容量为 8，则当您首次设置暖池并且该池正在初始化时，该暖池的大小将为 2。

在使用较大的自动扩缩组来管理使用暖池的成本效益时，您可能仅需要使用最大预热容量选项。例如，与在暖池中持续预留 500 个实例以备未来使用的方案相比，拥有 1,000 个实例的自动扩缩组、最大容量为 1,500（用于为紧急流量峰值提供额外容量）和拥有 100 个实例的暖池方案可能有利于更好地实现目标。

最小暖池大小

考虑使用最小大小设置来静态设置要在暖池中保留的最少实例数。默认情况下，没有设置最小大小。

暖池实例状态

您可以将暖池中的实例保持在以下三种状态之一：`Stopped`、`Running` 或 `Hibernated`。将实例保持在 `Stopped` 状态是一个有效的方法，以最大限度地降低成本。对于已停止的实例，您只需为您使用的卷和已附加到实例的弹性 IP 地址付费。

或者，您也可以将实例保持在 `Hibernated` 状态，以停止实例而不删除其内存内容 (RAM)。当某个实例处于休眠状态时，这表示操作系统会将 RAM 的内容保存到 Amazon EBS 根卷中。当再次开启此实

例时，会将此根卷还原到其之前的状态，还将重新加载 RAM 内容。当实例处于休眠状态时，您只需为 EBS 卷（包括 RAM 内容的存储空间，以及已附加到实例的弹性 IP 地址）付费。

此外还可以将暖池中的实例保持在 `Running` 状态，但我们强烈建议不要这样操作，以避免产生不必要的费用。当实例停止或休眠时，您可以节省实例本身的成本。您仅在实例正在运行时为它们付费。

生命周期钩子

生命周期钩子 (p. 191)允许您将实例置于等待状态，以便您可以对实例执行自定义操作。自定义操作将在实例启动时或其终止之前执行。

在暖池配置中，生命周期钩子还可以延迟实例停止或休眠以及延迟在横向扩展事件期间投入使用，直到它们完成初始化。如果您在没有生命周期钩子的情况下向 Auto Scaling 组添加暖池，则需要很长时间才能完成初始化的实例可能会停止或休眠，然后在准备就绪之前在横向扩展事件期间投入使用。

实例再使用策略

默认情况下，Amazon EC2 Auto Scaling 会在您的 Auto Scaling 组横向缩减时终止您的实例。然后，它会将新实例启动到暖池中，以替换已终止的实例。

而如果您希望将实例返回到暖池，则可以指定实例再使用策略。这使您可以再使用这些已配置为给应用程序提供流量的实例。为了确保您的暖池不会过度预置，当暖池中的实例大小根据其设置超出必要时，Amazon EC2 Auto Scaling 可以终止这些实例，以减小其大小。当终止暖池中的实例时，它会使用默认终止策略 (p. 224)来选择首先终止哪些实例。

Important

如果您希望在横向缩减时休眠实例，并且 Auto Scaling 组中有现有实例，则它们必须满足实例休眠的要求。如果不这样做，当实例返回到暖池时，它们将回退到停止状态，而不是休眠状态。

Note

目前，您只能通过使用 Amazon CLI 或 SDK 来指定实例再使用策略。此功能在控制台中不可用。

先决条件

决定如何使用生命周期钩子来准备实例以供使用。可以通过两种方法对您的实例执行自定义操作。

- 对于您希望在启动时对实例运行命令的简单场景，可以在为 Auto Scaling 组创建启动模板或启动配置时包含用户数据脚本。用户数据脚本只是在您的实例开启时由 cloud-init 运行的普通 shell 脚本或 cloud-init 指令。该脚本还可通过使用实例（该脚本运行所在的实例）的 ID 来控制您的实例何时转换到下一个状态。如果您尚未这样做，请更新脚本，从实例元数据中检索实例的实例 ID。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的检索实例元数据。

  Tip

  要在实例重新开启时运行用户数据脚本，用户数据必须采用 MIME 多段格式，并在用户数据的 `#cloud-config` 部分中指定以下内容：

  ```
  #cloud-config
  cloud_final_modules:
   - [scripts-user, always]
  ```

- 对于需要服务（例如 Amazon Lambda 要在实例进入或离开暖池时执行某些操作）的高级场景，您可以为 Auto Scaling 组创建生命周期钩子，并将目标服务配置为根据生命周期通知执行自定义操作。有关更多信息，请参阅支持的通知目标 (p. 218)。

有关更多信息，请参阅我们的 GitHub 存储库中的生命周期钩子示例。

使实例为休眠做好准备

要使 Auto Scaling 实例为使用 `Hibernated` 池状态做好准备，请按照适用于 Linux 实例的 Amazon EC2 用户指南中的休眠先决条件主题中所述，创建正确设置的新启动模板或启动配置，以支持实例休眠。然后，将新启动模板或启动配置与 Auto Scaling 组关联起来，并开启实例刷新，以替换与之前的启动模板或启动配置关联的实例。有关更多信息，请参阅基于实例刷新替换 Auto Scaling 实例 (p. 95)。

创建暖池

按照以下说明使用控制台创建暖池。

在您开始之前，请确认您已为 Auto Scaling 组创建了生命周期钩子。

创建暖池（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中现有组旁边的复选框。

   将在 Auto Scaling group（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 选择实例管理选项卡。
4. 在暖池下，选择创建暖池。
5. 要配置暖池，请执行以下操作：

   a. 对于暖池实例状态，选择要在实例进入暖池时将其转换为哪个状态。默认为 `Stopped`。
   b. 对于最小暖池大小，输入要在暖池中保留的最少实例数。
   c. 对于 Max prepared capacity（最大准备容量），您可以通过定义一组固定数量的实例来指定最大准备容量，或者保留原定设置选项以不定义最大准备容量。

      如果保留原定设置 Equal to the Auto Scaling group's maximum capacity（等于 Auto Scaling 组的最大容量），则暖池的大小将与 Auto Scaling 组的最大容量与所需容量之差一致。要通过调整组的最大容量来更轻松地管理暖池的大小，我们建议您使用原定设置选项。

      如果选择 Define a set number of instances（定义一组固定数量的实例）选项，请输入一个表示允许同时位于暖池和 Auto Scaling 组中的最大实例数的值。
6. 选择 Create（创建）。

更新暖池

要更改暖池的启动模板或启动配置，请将新启动模板或启动配置与 Auto Scaling 组关联起来。所有新实例都将使用新 AMI 和在启动模板或启动配置中指定的其他更新启动，但现有实例不受影响。

要强制启动使用新启动模板或启动配置的替换暖池实例，您可以终止暖池中的现有实例。Amazon EC2 Auto Scaling 立即开始启动新实例以替换您终止的实例。或者，您可以启动实例刷新以执行组的滚动更新。实例刷新首先将替换 `InService` 实例。然后，它将替换暖池中的实例。有关更多信息，请参阅基于实例刷新替换 Auto Scaling 实例 (p. 95)。

删除暖池

当您不再需要暖池时，您可以使用以下步骤将其删除。

要删除暖池（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。

2. 选中现有组旁边的复选框。

   将在 Auto Scaling group（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 选择实例管理选项卡。
4. 对于 Warm pool（暖池），选择 Actions（操作）、Delete（删除）。
5. 当系统提示进行确认时，选择 Delete（删除）。

限制

- 您不能将暖池添加到具有混合实例策略或启动 Spot 实例的 Auto Scaling 组。
- 仅当 Amazon EC2 Auto Scaling 将 Amazon EBS 卷作为其根设备时，前者才可将实例置于 `Stopped` 或 `Hibernated` 状态。无法停止或休眠对根设备使用实例存储的实例。
- 仅当满足适用于 Linux 实例的 Amazon EC2 用户指南内的休眠先决条件主题中列出的所有要求时，Amazon EC2 Auto Scaling 才可将实例置于 `Hibernated` 状态。
- 如果您的暖池在发生向外扩展事件时耗尽，则实例将直接启动到 Auto Scaling 组（冷启动）。如果可用区容量不足，您也可能会遇到冷启动。
- 如果您尝试将暖池与 Amazon Elastic Kubernetes Service (Amazon EKS) 托管节点组一起使用，则仍在初始化的实例可能会注册到您的 Amazon EKS 集群。因此，在准备停止或休眠时，集群可能会在实例上安排作业。
- 同样，如果您尝试将暖池与 Amazon ECS 集群一起使用，则实例可能会在完成初始化之前注册到集群。要解决此问题，您必须配置启动模板或启动配置，使其包含用户数据中的特殊代理配置变量。有关更多信息，请参阅《Amazon Elastic Container Service 开发人员指南》中的为自动扩缩组使用暖池。
- 对暖池的休眠支持已在所有提供 Amazon EC2 Auto Scaling 的所有商业 Amazon Web Services 区域 开放，但中东（阿联酋）、中国（北京）、中国（宁夏）、和 Amazon GovCloud（美国东部和美国西部）区域除外。

将生命周期钩子与暖池一起使用

暖池中的实例将保持自己独立的生命周期，以帮助您为每次转换创建适当的自定义操作。此生命周期旨在帮助您在实例仍在初始化之时以及将其投入使用之前，调用目标服务（例如，Lambda 函数）中的操作。

> **Note**
>
> 不会更改用于添加和管理生命周期钩子以及完成生命周期操作的 API 操作。只会更改实例生命周期。

有关添加生命周期钩子的更多信息，请参阅 添加生命周期钩子 (p. 199)。有关完成生命周期操作的更多信息，请参阅 完成生命周期操作 (p. 201)。

对于进入暖池的实例，出于以下原因之一，您可能需要生命周期钩子：

- 您希望从 AMI 启动 EC2 实例，这需要很长时间才能完成初始化。
- 您希望运行用户数据脚本来引导启动 EC2 实例。

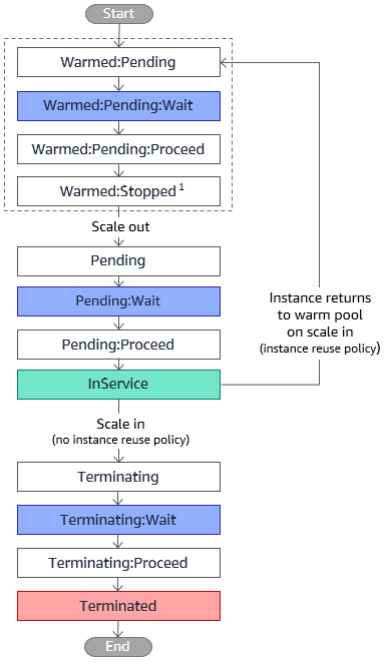
对于离开暖池的实例，出于以下原因之一，您可能需要生命周期钩子：

- 您可以使用一些额外的时间准备 EC2 实例以供使用。例如，您可能有这样的服务，它们必须在实例重新开启时开启，然后您的应用程序才能正常工作。
- 您希望预填充缓存数据，以使新服务器不会使用空缓存启动。
- 您希望使用配置管理服务将新实例注册为托管实例。

暖池中实例的生命周期状态转换

作为其生命周期的组成部分，Auto Scaling 实例可在多种状态之间转换。

下图显示了在您使用暖池时 Auto Scaling 状态之间的转换：

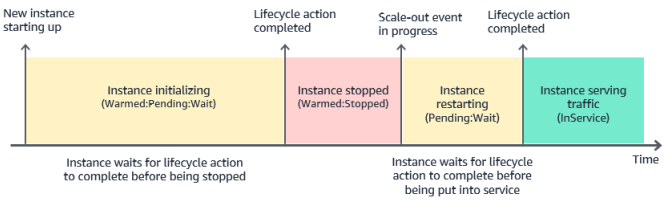


[1] 此状态因暖池的池状态设置而异。如果将池状态设置为 `Running`，则此状态为 `Warmed:Running`。如果将池状态设置为 `Hibernated`，则此状态为 `Warmed:Hibernated`。

添加生命周期钩子时，请考虑以下事项：

- 为 `autoscaling:EC2_INSTANCE_LAUNCHING` 生命周期操作配置生命周期挂钩时，新启动的实例在到达 `Warmed:Pending:Wait` 状态时会先暂停以执行自定义操作，然后在实例重启并到达 `Pending:Wait` 状态时再次重复上述操作。
- 为 `EC2_INSTANCE_TERMINATING` 生命周期操作配置生命周期挂钩，终止的实例在到达 `Terminating:Wait` 状态时会暂停以执行自定义操作。但是，如果您指定了实例重用策略来将实例大规模返回到暖池，而不是终止它们，则对于 `EC2_INSTANCE_TERMINATING` 生命周期操作，返回暖池的实例将在 `Warmed:Pending:Wait` 状态暂停以执行自定义操作。
- 如果对您的应用程序的需求耗尽了暖池，则 Amazon EC2 Auto Scaling 可以直接将实例启动到 Auto Scaling 组，前提条件是该组尚未达到其最大容量。如果实例直接启动到该组中，则这些实例仅在 `Pending:Wait` 状态中暂停以执行自定义奥做。
- 要控制实例在转换到下一个状态之前保持等待状态的时间，请将您的自定义操作配置为使用 complete-lifecycle-action 命令。使用生命周期挂钩，实例将保持等待状态，直到您通知 Amazon EC2 Auto Scaling 指定的生命周期操作已完成，或者直至超时期结束（默认为 1 小时）。

以下总结了横向扩展事件的流程。



当实例达到等待状态时，Amazon EC2 Auto Scaling 将发送通知。这些通知的示例可在本指南的 EventBridge 部分中找到。有关更多信息，请参阅暖池活动类型和模式 (p. 293)。

支持的通知目标

Amazon EC2 Auto Scaling 支持将以下任意内容定义为生命周期通知的通知目标：

- EventBridge 规则
- Amazon SNS 主题
- Amazon SQS 队列

> **Important**
>
> 请记住，如果您的启动模板或启动配置中有在实例启动时配置实例的用户数据 (cloud-init) 脚本，则无需接收通知即可对正在启动或重启的实例执行自定义操作。

以下各节包含多个链接，它们指向描述如何配置通知目标的文档：

EventBridge 规则：要在 Amazon EC2 Auto Scaling 将实例置于等待状态时运行代码，您可以创建 EventBridge 规则，并指定 Lambda 函数作为其目标。要根据不同的生命周期通知调用不同的 Lambda 函数，您可以创建多条规则，并将每条规则与特定的事件模式和 Lambda 函数关联起来。有关更多信息，请参阅为暖池事件创建 EventBridge 规则 (p. 299)。

Amazon SNS 主题：要在实例处于等待状态时接收通知，您可以创建 Amazon SNS 主题，然后设置 Amazon SNS 消息筛选，以根据消息属性以不同的方式传送生命周期通知。有关更多信息，请参阅使用 Amazon SNS 接收通知 (p. 196)。

Amazon SQS 队列：要为生命周期通知设置传送点，使相关使用者可以接收并处理它们，您可以创建 Amazon SQS 队列和处理来自 SQS 队列的消息的队列使用者。如果您希望队列使用者根据消息属性以不同方式处理生命周期通知，则还必须设置队列使用者以解析消息，然后在特定属性与所需值匹配时对消息执行操作。有关更多信息，请参阅使用 Amazon SQS 接收通知 (p. 197)。

查看运行状况检查状态以及运行状况检查失败的原因

运行健康状况检查允许 Amazon EC2 Auto Scaling 确定实例何时运行状况不佳并应终止。对于保持在 `Stopped` 状态的暖池实例，它采用了 Amazon EBS 拥有的 `Stopped` 实例的可用性知识来识别运行状况不佳的实例。它通过调用 `DescribeVolumeStatus` API 以确定附加到实例的 EBS 卷的状态来执行此操作。对于保存在 `Running` 状态的暖池实例，它依赖 EC2 状态检查来确定实例运行状况。虽然暖池实例没有运行状况检查宽限期，但 Amazon EC2 Auto Scaling 在生命周期钩子结束之前不会开始检查实例运行状况。

当发现实例运行状况不佳时，Amazon EC2 Auto Scaling 会自动删除运行状况不佳的实例，并创建一个新实例来替换它。实例通常在运行状况检查失败后几分钟内终止。有关更多信息，请参阅替换运行不正常的实例 (p. 244)。

还支持自定义运行状况检查。如果您拥有自己的运行状况检查系统，该系统可以检测实例的运行状况并将此信息发送到 Amazon EC2 Auto Scaling，则这将非常有用。有关更多信息，请参阅自定义运行状况检测 (p. 244)。

在 Amazon EC2 Auto Scaling 控制台中，您可以查看暖池实例的状态（运行状况良好或运行状况不佳）。您也可以使用 Amazon CLI 或其中一个 SDK 来查看其运行状况。

查看暖池实例的状态（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。

3. 在实例管理选项卡的暖池实例下，生命周期列显示实例的状态。

   运行状况列显示了 Amazon EC2 Auto Scaling 对实例运行状况所做的评估。

   Note

   新实例开始运行状况良好。在生命周期钩子完成之前，不会检查实例的运行状况。

查看运行状况检查失败原因（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在活动选项卡的活动历史记录下，状态列显示您的 Auto Scaling 组是否已成功启动或终止实例。

   如果它终止了任何运行状况不佳的实例，原因列显示终止的日期和时间以及运行状况检查失败的原因。例如，“在 2021-04-01T21:48:35Z 时，实例因 EBS 卷运行状况检查失败而停止服务”。

查看暖池实例的状态 (Amazon CLI)

通过使用以下 describe-warm-pool 命令查看 Auto Scaling 组的暖池。

```
aws autoscaling describe-warm-pool --auto-scaling-group-name my-asg
```

输出示例。

```
{
    "WarmPoolConfiguration": {
        "MinSize": 0,
        "PoolState": "Stopped"
    },
    "Instances": [
        {
            "InstanceId": "i-0b5e5e7521cfaa46c",
            "InstanceType": "t2.micro",
            "AvailabilityZone": "us-west-2a",
            "LifecycleState": "Warmed:Stopped",
            "HealthStatus": "Healthy",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-08c4cd42f320d5dcd",
                "LaunchTemplateName": "my-template-for-auto-scaling",
                "Version": "1"
            }
        },
        {
            "InstanceId": "i-0e21af9dcfb7aa6bf",
            "InstanceType": "t2.micro",
            "AvailabilityZone": "us-west-2a",
            "LifecycleState": "Warmed:Stopped",
            "HealthStatus": "Healthy",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-08c4cd42f320d5dcd",
                "LaunchTemplateName": "my-template-for-auto-scaling",
                "Version": "1"
            }
        }
    ]
}
```

查看运行状况检查失败原因 (Amazon CLI)

使用以下 describe-scaling-activities 命令。

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

以下为示例响应，其中 Description 表示您的 Auto Scaling 组已终止实例，Cause 指示运行状况检查失败的原因。

扩展活动按开始时间排序。首先描述仍在进行的活动。

```
{
  "Activities": [
    {
      "ActivityId": "4c65e23d-a35a-4e7d-b6e4-2eaa8753dc12",
      "AutoScalingGroupName": "my-asg",
      "Description": "Terminating EC2 instance: i-04925c838b6438f14",
      "Cause": "At 2021-04-01T21:48:35Z an instance was taken out of service in response to
 EBS volume health check failure.",
      "StartTime": "2021-04-01T21:48:35.859Z",
      "EndTime": "2021-04-01T21:49:18Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-west-2a
\"...}",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    },
...
  ]
}
```

使用 Amazon CLI 创建和管理暖池的示例

您可以使用 Amazon Web Services Management Console、Amazon Command Line Interface (Amazon CLI) 或 SDK 创建和管理暖池。

以下示例向您演示如何使用 Amazon CLI 创建和管理暖池。

目录

- 示例 1：将实例保持在 Stopped 状态 (p. 220)
- 示例 2：将实例保持在 Running 状态 (p. 221)
- 示例 3：将实例保持在 Hibernated 状态 (p. 221)
- 示例 4：在横向缩减时将实例返回到暖池 (p. 221)
- 示例 5：指定暖池中的最小实例数 (p. 221)
- 示例 6：单独定义暖池的最大容量 (p. 221)
- 示例 7：定义绝对暖池大小 (p. 221)
- 示例 8：删除暖池 (p. 222)

示例 1：将实例保持在 Stopped 状态

以下 put-warm-pool 示例创建一个暖池，它使实例保持在 Stopped 状态。

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped
```

示例 2：将实例保持在 Running 状态

以下 put-warm-pool 示例创建一个暖池，它使实例保持在 Running 状态而非 Stopped 状态。

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Running
```

示例 3：将实例保持在 Hibernated 状态

以下 put-warm-pool 示例创建一个暖池，它使实例保持在 Hibernated 状态而非 Stopped 状态。这使您可以停止实例，而无需删除其内存内容 (RAM)。

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Hibernated
```

示例 4：在横向缩减时将实例返回到暖池

以下 put-warm-pool 示例创建一个暖池，它使实例保持在 Stopped 状态，并包括 --instance-reuse-policy 选项。实例再使用策略值 '{"ReuseOnScaleIn": true}' 将在 Auto Scaling 组横向缩减时，通知 Amazon EC2 Auto Scaling 将实例返回到暖池。

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --instance-reuse-policy '{"ReuseOnScaleIn": true}'
```

示例 5：指定暖池中的最小实例数

以下 put-warm-pool 示例会创建一个暖池，它维护至少 4 个实例，以便至少有 4 个实例可用于处理流量峰值。

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --min-size 4
```

示例 6：单独定义暖池的最大容量

一般来说，您了解需要将最大容量设置为超过所需容量多少。通常无需定义额外的最大容量，因为 Amazon EC2 Auto Scaling 会创建一个将根据该组所需的容量和最大容量动态调整容量的暖池。但您可以在需要时使用 --max-group-prepared-capacity 选项单独定义暖池的最大容量。

以下 put-warm-pool 示例创建了一个单独定义最大容量的个暖池。假定 Auto Scaling 组的所需容量为 800。当您运行此命令并且池正在初始化时，暖池的大小将为 100。

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --max-group-prepared-capacity 900
```

要在暖池中保留最少数量的实例，请使用命令包含 --min-size 选项，如下所示。

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --max-group-prepared-capacity 900 --min-size 25
```

示例 7：定义绝对暖池大小

如果您为 --max-group-prepared-capacity 和 --min-size 选项设置了相同的值，则暖池将具有绝对大小。以下 put-warm-pool 示例创建一个暖池，它将保持 10 个实例的恒定暖池大小。

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /
  --pool-state Stopped --min-size 10 --max-group-prepared-capacity 10
```

示例 8：删除暖池

使用以下 delete-warm-pool 命令删除暖池。

```
aws autoscaling delete-warm-pool --auto-scaling-group-name my-asg
```

如果暖池中存在实例，或者正在进行扩展活动，请使用删除暖池命令与 `--force-delete` 选项。此选项还将终止 Amazon EC2 实例和任何未完成的生命周期操作。

```
aws autoscaling delete-warm-pool --auto-scaling-group-name my-asg --force-delete
```

控制在横向缩减过程中要终止的 Auto Scaling 实例

Amazon EC2 Auto Scaling 使用终止策略来确定它在扩展事件期间首先终止哪些实例。终止策略定义了 Amazon EC2 Auto Scaling 在选择要终止的实例时使用的终止条件。

Auto Scaling 组使用默认终止策略，但您可以根据自己的终止条件选择或创建自己的终止策略。这样，您就可以确保根据您的特定应用程序需求终止您的实例。

Amazon EC2 Auto Scaling 还提供了实例缩减保护。启用此功能后，它会防止在扩展事件期间终止实例。您可以在创建 Auto Scaling 组时启用实例缩减保护，并可以更改运行的实例上的设置。如果您在现有 Auto Scaling 组上启用实例缩减保护，则在此之后启动的所有新实例都将启用实例缩减保护。

> **Note**
>
> 实例扩展保护不能保证在发生人为错误时不会终止实例，例如，有人使用 Amazon EC2 控制台或 Amazon CLI 手动终止。要防止您的实例被意外终止，可使用 Amazon EC2 终止保护。但是，即使启用了终止保护和实例扩展保护，如果运行状况检查确定实例运行状况不佳或组本身被意外删除，保存到实例存储的数据也可能会丢失。与任何环境一样，最佳做法是频繁备份您的数据，或者在适合您的业务连续性要求时备份数据。

目录

- 终止策略使用方案 (p. 222)
- 使用 Amazon EC2 Auto Scaling 终止策略 (p. 224)
- 了解使用 Lambda 创建自定义终止策略。 (p. 227)
- 实例横向缩减保护 (p. 230)

终止策略使用方案

下面的部分介绍 Amazon EC2 Auto Scaling 使用终止策略的情形。

目录

- 缩减事件 (p. 222)
- 实例刷新 (p. 223)
- 重新平衡可用区 (p. 223)

缩减事件

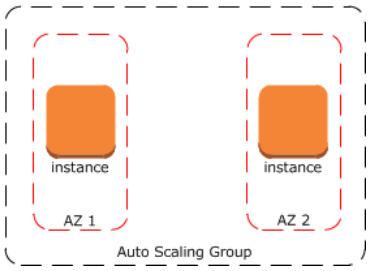
当 Auto Scaling 组的所需容量有一个低于该组当前容量的新值时，就会发生放大事件。

在以下情况下发生扩展事件：

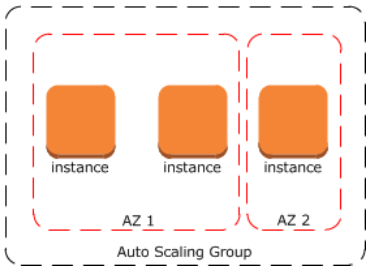
- 使用动态扩展策略时，组的大小会因指标值的更改而减小
- 当使用计划的扩展时，组的大小由于计划的操作而减小
- 当手动减小组的大小

以下示例显示了在发生扩展事件时终止策略的工作原理。

1. 例如，假设您有一个包含一个实例类型、两个可用区和两个实例所需容量的 Auto Scaling group 组。它还具有一个动态扩展策略，可在资源利用率增加或减少时添加和删除实例。该组中的两个实例分布在两个可用区之间，如下图所示。



2. 当 Auto Scaling 组扩展时，Amazon EC2 Auto Scaling 会启动一个新实例。Auto Scaling 组现在有三个实例，分布在两个可用区，如下图所示。



3. 当 Auto Scaling 组扩展时，Amazon EC2 Auto Scaling 将终止其中一个实例。
4. 如果您没有为该组分配特定的终止策略，则 Amazon EC2 Auto Scaling 使用默认终止策略。该策略会选择具有两个实例的可用区，并终止从最老的启动木板或启动配置启动的实例。如果这些实例是从同一启动模板或启动配置启动的，Amazon EC2 Auto Scaling 会选择终止最接近下个计费小时的实例。

实例刷新

您可以启动实例刷新以更新 Auto Scaling 组中的实例。在实例刷新期间，Amazon EC2 Auto Scaling 会终止组中的实例，并启动已终止实例的替换实例。Auto Scaling 组的终止策略控制首先替换哪些实例。

重新平衡可用区

Amazon EC2 Auto Scaling 在为 Auto Scaling 组启用的可用区之间均匀平衡您的容量。这有助于减少可用区中断的影响。如果跨可用区域的容量分配不平衡，Amazon EC2 Auto Scaling 会重新平衡 Auto Scaling 组，方法是在具有最少实例的已启用可用区中启动实例，并在其他地方终止实例。终止策略控制哪些实例优先终止。

跨可用区的实例分布可能失去平衡的原因有很多。

移除实例

> 如果您从 Auto Scaling 组中分离实例，将实例置于备用状态或者明确终止实例并减少所需容量，从而阻止启动替换实例，则该组可能会变得不平衡。若出现这种情况，Amazon EC2 Auto Scaling 将通过再平衡可用区来进行补偿。

使用不同于最初指定的可用区

如果您扩展 Auto Scaling 组以包含更多可用区，或者更改所使用的可用区，则 Amazon EC2 Auto Scaling 会在新可用区中启动实例并终止其他可用区中的实例，以帮助确保实例在各个可用区中均匀分布。

可用性中断

可用性中断的情况很少发生。但是，如果某个可用区变得不可用并稍后恢复，则您的 Auto Scaling 组可能会在可用区之间变得不均衡。Amazon EC2 Auto Scaling 会尝试逐步再平衡该组，而该再平衡操作可能会终止其他区中的实例。

例如，假设您有一个包含一个实例类型、两个可用区和两个实例所需容量的 Auto Scaling 组。当一个可用区出现故障时，Amazon EC2 Auto Scaling 会在运行状况良好的可用区中自动启动一个新实例，以替换运行状况不佳的可用区中的实例。当运行状况不佳的可用区恢复到正常状态时，Amazon EC2 Auto Scaling 会自动在此可用区中启动一个新实例，同时终止未受影响的可用区中的实例。

Note

再平衡时，Amazon EC2 Auto Scaling 在终止旧实例之前启动新实例，所以再平衡不会损害应用程序的性能或可用性。
因为 Amazon EC2 Auto Scaling 尝试在终止旧实例之前启动新实例，所以达到或接近指定的最大容量可能会阻碍或完全停止再平衡活动。为避免此问题，在再平衡活动期间，系统可以暂时超出某组的指定最大容量的 10%（或 1 个实例边缘，以较大者为准）。仅当该组达到或接近最大容量，并需要重新平衡时，才可超出容量限制；此类情况的原因是用户请求重新分区，或者是为了弥补区域可用性问题。该超出状态仅持续重新平衡该组所需的时间。

使用 Amazon EC2 Auto Scaling 终止策略

本主题提供有关默认终止策略的详细信息，以及可供您为个别 Auto Scaling 组选择不同终止策略的选项。通过选择不同的终止策略，您可以控制在发生缩减事件时首先终止哪些实例。例如，您可以选择不同的终止策略，以便 Amazon EC2 Auto Scaling 先终止最旧实例的优先级。

当 Amazon EC2 Auto Scaling 终止实例时，它将尝试在 Auto Scaling 组使用的可用区之间保持均衡。保持跨可用区的平衡优先于终止策略。如果一个可用性区域的实例多于该组使用的其他可用性区域，Amazon EC2 Auto Scaling 会将终止策略应用于不平衡可用性区域的实例。如果组使用的可用区是均衡的，则 Amazon EC2 Auto Scaling 将应用于该组的所有可用区。

目录

- 默认终止策略 (p. 224)
- 默认终止策略和混合实例组 (p. 225)
- 使用不同的终止策略 (p. 225)
  - 使用不同的终止策略（控制台）(p. 226)
  - 使用不同的终止策略（Amazon CLI）(p. 226)

默认终止策略

在选择要终止的实例之前，默认终止策略会应用多个终止条件。当 Amazon EC2 Auto Scaling 终止实例时，它首先确定哪些可用性区域具有最多的实例，并找到至少一个不受扩展保护的实例。在选定的可用区内，以下默认终止策略行为适用：

1. 确定是否任何有终止资格的实例使用最旧的启动模板或启动配置：
   a. [对于使用启动模板的 Auto Scaling 组]

      确定任何实例是否使用最早的启动配置或模板，除非有使用启动配置的实例。Amazon EC2 Auto Scaling 先终止使用启动配置的实例，然后才终止使用启动模板的实例。

b. [对于使用启动配置的 Auto Scaling 组]

确定是否有任何实例使用最旧的启动配置。

2. 在应用上述条件后，如果要终止多个不受保护的实例，请确定哪些实例最接近下一个计费小时。如果有多个不受保护的实例最接近下一个计费小时，请随机终止其中的一个实例。

请注意，通过终止最接近下一个计费小时的实例，将帮助您最大程度地使用按小时计费的实例。或者，如果您的 Auto Scaling 组使用 Amazon Linux、Windows 或 Ubuntu，则 EC2 的使用以秒为单位计费。有关更多信息，请参阅 Amazon EC2定价。

默认终止策略和混合实例组

当 Auto Scaling 组具有混合实例策略 (p. 50)扩展时，Amazon EC2 Auto Scaling 仍然使用终止策略来确定要终止的实例的优先级，但首先确定应终止两种类型中的哪一种类型（竞价型或按需型）。然后，它将在每个可用区中分别应用终止策略。它还可以识别哪些实例（在已确定的购买选项内），其中可用区将导致可用区域最平衡。同样的逻辑适用于使用混合实例配置并为实例类型定义了权重的 Auto Scaling 组。

默认终止策略略略有变化，原因是混合实例策略 (p. 50)实现。默认终止策略的以下新行为适用：

1. 确定要终止的实例，以便使剩余实例与要终止的按需实例或竞价型实例的分配策略 (p. 50)保持一致。

例如，在实例启动后，您将更改首选实例类型的优先级顺序。发生缩减事件时，Amazon EC2 Auto Scaling 尝试从优先级较低的实例类型逐渐转移按需实例。

2. 确定任何实例是否使用最早的启动配置或模板，除非有使用启动配置的实例。Amazon EC2 Auto Scaling 先终止使用启动配置的实例，然后才终止使用启动模板的实例。
3. 在应用上述条件后，如果要终止多个不受保护的实例，请确定哪些实例最接近下一个计费小时。如果有多个不受保护的实例最接近下一个计费小时，请随机终止其中的一个实例。

使用不同的终止策略

要指定要在 Amazon EC2 Auto Scaling 选择终止实例之前应用的终止条件，您可以从以下任何预定义的终止策略中进行选择：

- `Default`。根据默认终止策略终止实例。当您希望在任何其他策略之前评估竞价分配策略时，此策略非常有用，因此每次终止或替换您的竞价型实例时，您都可以继续使用最佳池中的竞价型实例。当您想要移除启动配置并开始使用启动模板时，它也很有用。
- `AllocationStrategy`。终止 Auto Scaling 组中的实例，使剩余实例与所终止实例类型的分配策略匹配（竞价型实例或按需型实例）。当您首选的实例类型发生变化时，可使用该策略。如果竞价分配策略是 `lowest-price`，则您可以逐渐在 N 个最低价竞价池中重新平衡分布竞价型 实例。如果竞价分配策略是 `capacity-optimized`，则您可以逐渐重新平衡具有更多可用竞价容量的竞价池之间 竞价型实例的分布。您也可以使用较高优先级类型的按需实例逐渐替代较低优先级类型的按需实例。
- `OldestLaunchTemplate`。终止采用最旧启动模板的实例。利用此策略，会首先终止使用非当前启动模板的实例，然后终止使用当前启动模板的最旧版本的实例。如果要更新某个组并且逐步淘汰先前配置中的实例时，此策略非常有用。
- `OldestLaunchConfiguration`。终止采用最旧启动配置的实例。如果要更新某个组并且逐步淘汰先前配置中的实例时，此策略非常有用。使用此策略，使用非当前启动配置的实例将首先终止。
- `ClosestToNextInstanceHour`。终止最接近下个计费小时的实例。此策略有助于最大限度使用按小时收费的实例。（只有使用 Amazon Linux、Windows 或 Ubuntu 的实例以秒为单位计费。）
- `NewestInstance`。终止组中最新的实例。如果要测试新的启动配置但不想在生产中保留它时，此策略非常有用。
- `OldestInstance`。终止组中最旧的实例。当您将 Auto Scaling 组中的实例升级为新的 EC2 实例类型，此选项十分有用。您可以逐渐将较旧类型的实例替换为较新类型的实例。

Note

Amazon EC2 Auto Scaling 始终首先在可用区间平衡实例，无论使用何种终止策略。因此，您可能会遇到一些较新的实例在旧实例之前终止的情况。例如，当某个可用区的实例数多于该组使用的其他可用区时，或某个可用区的实例数多于该组使用的其他可用区。

使用不同的终止策略（控制台）

创建 Auto Scaling 组后，您可以更新组的终止策略。默认终止策略被自动使用。您可以选择将默认策略替换为不同的终止策略（例如 `OldestLaunchTemplate`）或按照应用顺序列出的多个终止策略。

要选择不同的终止策略

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details（详细信息）选项卡上，选择 Advanced configurations（高级配置）、Edit（编辑）。
4. 对于 Termination policies（终止策略），请选择一个或多个终止策略。如果您选择多个策略，请按照您希望评估策略的顺序将其列出。

   您还可选择 Custom termination policy（自定义终止策略），然后选择一个满足您需求的 Lambda 函数。如果您已经为 Lambda 函数创建了版本和别名，则可以从 Version/Alias（版本/别名）下拉列表选择版本或别名。要使用您的 Lambda 函数的未发布版本，请保留 Version/Alias（版本/别名）设置为默认值。有关更多信息，请参阅 了解使用 Lambda 创建自定义终止策略。 (p. 227)。

   Note

   使用多个策略时，必须正确设置它们的顺序：

   - 如果您使用 Default（默认）策略，它必须是列表中的最后一个策略。
   - 如果您使用 Custom termination policy（自定义终止策略），它必须是列表中的第一个策略。
5. 选择 Update（更新）。

使用不同的终止策略（Amazon CLI）

自动使用默认终止策略，除非指定了不同的策略。

要使用不同的终止策略

使用以下命令之一：

- create-auto-scaling-group
- update-auto-scaling-group

您可以单独使用终止策略，或者将它们合并到策略列表中。例如，使用以下命令更新 Auto Scaling 组以首先使用 `OldestLaunchConfiguration` 策略，然后使用 `ClosestToNextInstanceHour` 策略。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --termination-policies "OldestLaunchConfiguration" "ClosestToNextInstanceHour"
```

如果您使用 `Default` 终止策略，请将该策略设为终止策略列表中的最后一项。例如，`--termination-policies "OldestLaunchConfiguration" "Default"`。

要使用自定义终止策略，必须首先使用 Amazon Lambda 创建终止策略。要指定要用作终止策略的 Lambda 函数，请将其设为终止策略列表中的第一个函数。例如，`--termination-policies "arn:aws:lambda:us-west-2:123456789012:function:HelloFunction:prod" "OldestLaunchConfiguration"`。有关更多信息，请参阅了解使用 Lambda 创建自定义终止策略。 (p. 227)。

了解使用 Lambda 创建自定义终止策略。

Amazon EC2 Auto Scaling 使用终止策略来确定在减小 Auto Scaling 组大小时首先终止哪些实例的优先级（称为缩减）。Auto Scaling 组使用默认终止策略，但您可以选择或创建您自己的终止策略。有关选择预定义终止策略的更多信息，请参阅 使用 Amazon EC2 Auto Scaling 终止策略 (p. 224)。

在本主题中，您将了解如何使用 Amazon Lambda 函数创建自定义终止策略，Amazon EC2 Auto Scaling 在响应某些事件时调用该函数。您创建的 Lambda 函数处理 Amazon EC2 Auto Scaling 发送的输入数据中的信息，并返回准备终止的实例列表。

自定义终止策略可以更好地控制终止哪些实例以及何时终止。例如，当 Auto Scaling 组扩展时，Amazon EC2 Auto Scaling 无法确定是否存在不应中断的运行的工作负载。使用 Lambda 函数，您可以验证终止请求并等待工作负载完成后再将实例 ID 返回 Amazon EC2 Auto Scaling 以进行终止。

目录

- 输入数据 (p. 227)
- 响应数据 (p. 228)
- 使用自定义终止策略时的注意事项 (p. 229)
- 创建 Lambda 函数 (p. 229)
- 限制 (p. 230)

输入数据

Amazon EC2 Auto Scaling 为扩展事件生成 JSON 有效负载，当实例由于实例最大生命周期或实例刷新功能而终止时，也会生成此负载。它还为在跨可用区重新平衡组时可以启动的扩展事件生成 JSON 有效负载。

此负载包含有关 Amazon EC2 Auto Scaling 需要终止的容量、建议终止的实例列表以及启动终止的事件的信息。

以下是示例负载：

```
{
  "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-1:<account-
id>:autoScalingGroup:d4738357-2d40-4038-ae7e-b00ae0227003:autoScalingGroupName/my-asg",
  "AutoScalingGroupName": "my-asg",
  "CapacityToTerminate": [
    {
      "AvailabilityZone": "us-east-1b",
      "Capacity": 2,
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1b",
      "Capacity": 1,
      "InstanceMarketOption": "spot"
    },
    {
      "AvailabilityZone": "us-east-1c",
      "Capacity": 3,
      "InstanceMarketOption": "on-demand"
    }
  ],
```

```
  "Instances": [
    {
      "AvailabilityZone": "us-east-1b",
      "InstanceId": "i-0056faf8da3e1f75d",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1c",
      "InstanceId": "i-02e1c69383a3ed501",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1c",
      "InstanceId": "i-036bc44b6092c01c7",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    ...
  ],
  "Cause": "SCALE_IN"
}
```

有效负载包括 Auto Scaling 组的名称、其 Amazon Resource Name (ARN) 以及以下元素：

- `CapacityToTerminate` 描述了在给定可用区域内设置为终止的竞价或按需容量的大小。
- `Instances` 表示 Amazon EC2 Auto Scaling 根据 `CapacityToTerminate` 信息建议终止的实例。
- `Cause` 描述了导致终止的事件：`SCALE_IN`、`INSTANCE_REFRESH`、`MAX_INSTANCE_LIFETIME` 或者 `REBALANCE`。

以下信息概述了 Amazon EC2 Auto Scaling 如何生成`Instances`输入数据中：

- 当实例由于扩大事件和基于实例刷新的终止而终止时，保持可用区间的平衡优先。因此，如果某个可用区的实例数多于该组使用的其他可用区，则您的终止策略将应用于不均衡可用区中的实例。如果组使用的可用区是均衡的，则输入数据将包含组的所有可用区中的实例。
- 使用混合实例策略 (p. 50)，则根据您对每个购买选项的所需百分比，维持您的竞价和按需容量的平衡也将优先考虑。我们首先确定应终止两种类型中的哪种类型（竞价或按需）。然后，我们确定哪些实例（在已确定的购买选项内），我们可以终止哪些可用区，这些实例将导致可用区域最平衡。

响应数据

输入数据和响应数据协同工作，以缩小要终止的实例列表。

对于给定的输入，Lambda 函数的响应应类似于以下示例：

```
{
  "InstanceIDs": [
    "i-02e1c69383a3ed501",
    "i-036bc44b6092c01c7",
    ...
  ]
}
```

这些区域有：`InstanceIDs` 表示准备终止的实例。

或者，您可以返回准备终止的一组不同的实例，这些实例将覆盖输入数据中的实例。如果在您的 Lambda 函数被调用时没有任何实例可以终止，您也可以选择不返回任何实例。

没有准备好终止的实例时，Lambda 函数的响应应类似于以下示例：

```
{
  "InstanceIDs": [ ]
}
```

使用自定义终止策略时的注意事项

注意以下使用自定义终止策略时的注意事项：

- 首先在响应数据中返回实例并不能保证其终止。如果在调用 Lambda 函数时返回的实例数超过所需数量，则 Amazon EC2 Auto Scaling 会根据您为 Auto Scaling 组指定的其他终止策略评估每个实例。当存在多个终止策略时，它会尝试应用列表中的下一个终止策略，如果实例数量超过了终止所需的实例，则会转到下一个终止策略，依此类推。如果未指定其他终止策略，则使用默认终止策略来确定要终止的实例。
- 如果没有返回任何实例或您的 Lambda 函数超时，则 Amazon EC2 Auto Scaling 会等待很短的时间，然后再次调用您的函数。对于任何扩展事件，只要组的所需容量小于其当前容量，它就会继续尝试。例如，基于刷新的终止，它会持续尝试一个小时。之后，如果它继续无法终止任何实例，则实例刷新操作将失败。在实例生命周期最长的情况下，Amazon EC2 Auto Scaling 会继续尝试终止被确定为超过其最大生命周期的实例。
- 由于您的函数会重复重试，因此请确保在使用 Lambda 函数作为自定义终止策略之前测试并修复代码中的任何永久性错误。
- 如果您使用自己的要终止的实例列表覆盖输入数据，并且终止这些实例会使可用区失去平衡，则 Amazon EC2 Auto Scaling 会逐渐重新平衡跨可用区域的容量分配。首先，它调用你的 Lambda 函数，查看是否有准备终止的实例，以便它可以确定是否开始重新平衡。如果存在可以终止的实例，它会首先启动新实例。当实例完成启动后，它会检测到组的当前容量高于其所需容量，并启动扩展事件。

创建 Lambda 函数

首先创建 Lambda 函数，以便您可以在 Auto Scaling 组的终止策略中指定 Amazon Resource Name (ARN)。

要创建 Lambda 函数（控制台）

1. 打开 Lambda 控制台的 Functions（函数）页面。
2. 在屏幕顶部的导航栏中，选择您在创建 Auto Scaling 组时使用的同一区域。
3. 选择 Create function（创建函数），然后选择 Author from scratch（从头开始创作）。
4. 在基本信息下的函数名称中输入函数的名称。
5. 选择 Create function（创建函数）。返回到函数的代码和配置。
6. 让函数在控制台中保持打开状态，在函数代码，请将代码粘贴到编辑器中。
7. 选择 Deploy（部署）。
8. 或者，通过选择版本选项卡，然后选择发布新版本，创建 Lambda 函数的发布版本。要了解有关 Lambda 中的版本控制的详细信息，请参阅 Amazon Lambda 开发人员指南中的 Lambda 函数版本。
9. 如果您选择发布版本，在别名选项卡选择是否要将别名与此版本的 Lambda 函数关联。要了解有关 Lambda 中的别名的详细信息，请参阅 Amazon Lambda 开发人员指南中的 Lambda 函数别名。
10. 接下来，选择配置选项卡，然后 Permissions（权限）。
11. 向下滚动到基于资源的策略，然后选择添加权限。基于资源的策略用于向在策略中指定的委托人授予调用函数的权限。在这种情况下，委托人将是与 Auto Scaling 组关联的 Amazon EC2 Auto Scaling 服务关联角色。
12. 在策略语句部分中，配置您的权限：

    a. 选择 Amazon Web Services 账户。

b. 适用于委托人中，输入调用服务链接角色的 ARN，例如 **arn:aws:iam::<aws-account-id>:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling**。
c. 对于操作，选择 lambda:InvokeFunction。
d. 对于声明 ID，输入唯一的声明 ID，如 **AllowInvokeByAutoScaling**。
e. 选择 Save（保存）。

13. 在遵循以上说明操作后，继续作为下一步，在 Auto Scaling 组的终止策略中指定函数的 ARN。有关更多信息，请参阅使用不同的终止策略（控制台）(p. 226)。

Note

有关可用作开发 Lambda 函数的参考示例，请参阅 Amazon EC2 Auto Scaling 的 GitHub 存储库

限制

- 您只能在 Auto Scaling 组的终止策略中指定一个 Lambda 函数。如果指定了多个终止策略，则必须先指定 Lambda 函数。
- 您可以使用非限定 ARN（不带后缀）或具有版本或别名作为后缀的限定 ARN 来引用 Lambda 函数。如果使用了不合格的 ARN（例如 `function:my-function`），您的基于资源的策略必须在函数的未发布版本上创建。如果使用了合格的 ARN（例如 `function:my-function:1` 或者 `function:my-function:prod`），您的基于资源的策略必须在函数的未发布版本上创建。
- 您不能使用后缀为 `$LATEST` 的合格 ARN。如果您尝试添加引用后缀为 `$LATEST` 的限定 ARN 的自定义终止策略，它将导致出现错误。
- 输入数据中提供的实例数量限制为 30,000 个实例。如果可以终止的实例超过 30,000 个，则输入数据将包含 `"HasMoreInstances": true` 以指示返回的实例的最大数目。
- Lambda 函数的最长运行时间为两秒（2000 毫秒）。作为最佳实践，您应该根据预期运行时间设置 Lambda 函数的超时值。Lambda 函数的默认超时时间为三秒，但这可以减少。
- Amazon EC2 Auto Scaling 不会终止启用了实例扩展保护的实例。

实例横向缩减保护

要控制 Auto Scaling 组在缩减时是否可以终止特定实例，请使用实例缩减保护。您可以对 Auto Scaling 组或单个 Auto Scaling 实例启用实例缩减保护设置。当 Amazon EC2 Auto Scaling 组启动新的实例或将暖池中的示例移入 Auto Scaling 组时，该实例会继承 Auto Scaling 组的实例缩减保护设置。您可以随时更改 Auto Scaling 组或 Auto Scaling 实例的实例缩减保护设置。

当实例状态为 `InService` 时，实例缩减保护启动。当您分离受终止保护的实例时，其实例横向缩减保护设置就会失效。再次将实例挂载到组时，它会继承组的当前实例缩减保护设置。

实例缩减保护并不能针对以下情况保护 Auto Scaling 实例：

- 通过 Amazon EC2 控制台、`terminate-instances` 命令或 `TerminateInstances` 操作手动终止。要保护 Auto Scaling 实例不被手动终止，请启用 Amazon EC2 终止保护。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南 中的启用终止支持。
- 实例未通过运行状况检查的情况下的运行状况检查更换。有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。
- 竞价型实例中断。当竞价型实例的容量不再可用或 Spot 价格超过您的最高价时，将终止该实例。

任务

- 为组启用实例缩减保护 (p. 231)
- 修改组的实例缩减保护设置 (p. 231)

- 修改实例的实例缩减保护设置 (p. 232)

Note

如果缩减期间 Auto Scaling 组中的所有实例都受终止保护并且发生缩减事件，则该组的所需容量会减少。不过，Auto Scaling 组不能终止所需数量的实例，直到其实例缩减保护设置被禁用。在 Amazon Web Services Management Console 中，如果 Auto Scaling 组中的所有实例在发生横向缩减事件时都受到保护无法横向缩减，则适用于 Auto Scaling 组的 Activity history（活动历史记录）将包括以下消息：`Could not scale to desired capacity because all remaining instances are protected from scale-in.`。

为组启用实例缩减保护

您可以在创建 Auto Scaling 组时启用实例缩减保护。默认情况下，实例缩减保护处于禁用状态。

启用实例缩减保护（控制台）

创建 Auto Scaling 组时，在 Configure group size and scaling policies（配置组大小和扩展策略）页面的 Instance scale-in protection（实例缩减保护）下，选择 Enable instance scale-in protection（启用实例缩减保护）选项。

启用实例缩减保护 (Amazon CLI)

使用以下 create-auto-scaling-group 命令启用实例横向缩减保护：

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --new-instances-protected-from-scale-in ...
```

修改组的实例缩减保护设置

您可以启用或禁用 Auto Scaling 组的实例缩减保护设置。启用实例缩减保护设置后，启用该设置后启动的所有新实例都将启用实例缩减保护。除非您为每个实例单独启用实例缩减保护设置，否则之前启动的实例不会受到扩展保护。

更改组的实例缩减保护设置（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details（详细信息）选项卡上，选择 Advanced configurations（高级配置）、Edit（编辑）。
4. 对于 Instance scale-in protection（实例缩减保护），选择 Enable instance scale-in protection（启用实例缩减保护）。
5. 选择 Update（更新）。

更改组的实例缩减保护设置 (Amazon CLI)

使用以下 update-auto-scaling-group 命令可对指定的 Auto Scaling 组启用实例保护：

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --new-instances-protected-from-scale-in
```

使用以下命令为指定组禁用实例缩减保护。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --no-new-
instances-protected-from-scale-in
```

修改实例的实例缩减保护设置

默认情况下，实例从其 Auto Scaling 组获取其实例缩减保护设置。但是，您随时可以对实例启用或禁用实例缩减保护。

更改实例的实例缩减保护设置（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Instance management（实例管理）选项卡的 Instances（实例）中，选择实例。
4. 要启用实例缩减保护，请依次选择 Actions（操作）和 Set scale-in protection（设置缩减保护）。系统提示时，选择 Set scale-in protection（设置缩减保护）。
5. 要禁用实例缩减保护，请依次选择 Actions（操作）和 Remove scale-in protection（删除缩减保护）。系统提示时，选择 Remove scale-in protection（删除缩减保护）。

更改实例的实例缩减保护设置 (Amazon CLI)

使用以下 set-instance-protection 命令可对指定实例启用实例横向缩减保护：

```
aws autoscaling set-instance-protection --instance-ids i-5f2e8a0d --auto-scaling-group-
name my-asg --protected-from-scale-in
```

使用以下命令为指定实例禁用实例缩减保护。

```
aws autoscaling set-instance-protection --instance-ids i-5f2e8a0d --auto-scaling-group-
name my-asg --no-protected-from-scale-in
```

临时从 Auto Scaling 组中移除实例

您可以将处于 `InService` 状态的实例置于 `Standby` 状态，更新实例或排查实例问题，然后将实例恢复运行状态。处于备用状态的实例仍是 Auto Scaling 组的一部分，但它们不会主动处理负载均衡器流量。

此功能可帮助您停止和启动实例或重新启动实例，而无需担心 Amazon EC2 Auto Scaling 在运行状况检查过程中或在扩展事件期间终止实例。

例如，您可以随时更改启动模板或启动配置，以更改 Auto Scaling 组的 Amazon Machine Image (AMI)。Auto Scaling 组启动的任何后续实例都使用此 AMI。不过，Auto Scaling 组不会更新当前正在运行的实例。您可以终止这些实例并让 Amazon EC2 Auto Scaling 替换它们，或者使用实例刷新功能终止和替换实例。或者，您可以将实例置于备用状态，更新软件，然后将实例恢复运行。

从 Auto Scaling 组中分离实例类似于将实例置于备用状态。如果您想要管理独立 EC2 实例并可能终止它们，则分离实例可能会很有用。有关更多信息，请参阅 从 Auto Scaling 组中分离 EC2 实例 (p. 129)。

> **Important**
>
> 当您将实例置于备用状态时，Auto Scaling 组可能会在不同可用区之间变得不平衡。Amazon EC2 Auto Scaling 将通过再平衡可用区来进行补偿，除非您暂停 `AZRebalance` 进程。有关更多信息，请参阅 暂停和恢复 Auto Scaling 组的进程 (p. 236)。

目录

- 备用状态的工作方式 (p. 233)
- 处于备用状态的实例的运行状况 (p. 233)
- 使用移除删除实例（控制台） (p. 234)
- 临时移除实例 (Amazon CLI) (p. 234)

Important

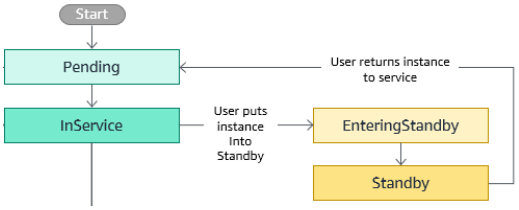
您需为处于备用状态的实例付费。

备用状态的工作方式

备用状态按如下方式工作以帮助您临时从 Auto Scaling 组中删除实例：

1. 将实例置于备用状态。实例保持此状态，直至您退出备用状态。
2. 如果已有负载均衡器目标组或经典负载均衡器附上 Auto Scaling 组，则会将实例取消注册到该负载均衡器。如果为负载均衡器启用了 Connection Draining，预设情况下，Elastic Load Balancing 会等待 300 秒，然后再完成注销过程，这有助于正在进行的请求的完成。
3. 在默认情况下，当您将一个实例置于备用状态时，指定作为所需容量的值会减少。这可防止在您将此实例置于备用状态时启动附加实例。或者，您可以指定不减少所需容量。如果您指定此选项，则 Auto Scaling 组会启动一个实例来替换处于备用状态的实例。这样做是为了帮助您在一个或多个实例处于备用状态时保持应用程序的容量。
4. 您可以更新实例或排查实例的问题。
5. 可通过退出备用状态，将实例恢复运行状态。
6. 在您将处于备用状态的实例恢复运行后，所需容量会增加。如果在将实例置于备用状态时未减少容量，Auto Scaling 组将检测到您具有超过所需数量的实例。它应用终止策略以实际减小组的大小。有关更多信息，请参阅 控制在横向缩减过程中要终止的 Auto Scaling 实例 (p. 222)。
7. 如果已有负载均衡器目标组或经典负载均衡器附上 Auto Scaling 组，则会将实例注册到该负载均衡器。

下图阐释了此过程中实例状态之间的过渡：



有关 Auto Scaling 组中实例的完整生命周期的更多信息，请参阅 Amazon EC2 Auto Scaling 实例生命周期 (p. 7)。

处于备用状态的实例的运行状况

Amazon EC2 Auto Scaling 不对处于备用状态的实例执行运行状况检查。当实例处于备用状态时，其运行状况将反映您将实例置于备用状态之前，实例具有的状态。Amazon EC2 Auto Scaling 不对实例执行运行状况检查，直至您将实例恢复运行。

例如，如果您将运行正常的实例置于备用状态，然后终止该实例，则 Amazon EC2 Auto Scaling 会继续将实例报告为运行正常。如果您尝试将处于备用状态的已终止实例重新投入使用，则 Amazon EC2 Auto Scaling 将对实例执行运行状况检查，确定其正在终止且运行状况不佳，并启动替换实例。有关运行状况检查的简介，请参阅 Auto Scaling 实例的运行状况检查 (p. 241)。

使用移除删除实例（控制台）

以下步骤演示更新当前正在运行的实例的一般过程。

临时删除实例

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Instance management（实例管理）选项卡的 Instances（实例）中，选择实例。
4. 选择 Actions（操作）和 Set to Standby（设置为备用）。
5. 在 Set to Standby（设置为备用）对话框上，选中该复选框可启动替代实例，取消选中可减少所需容量。选择 Set to Standby（设置为备用）。
6. 您可以根据需要更新实例或排查实例的问题。当您完成后，请继续下一步以将实例恢复运行。
7. 选择实例，再选择 Actions（操作）和 Set to InService（设置为可用）。在 Set to InService（设置为可用）对话框中，单击 Set to InService（设置为可用）。

临时移除实例 (Amazon CLI)

以下步骤演示更新当前正在运行的实例的一般过程。

临时删除实例

1. 使用下面的 describe-auto-scaling-instances 命令确定要更新的实例：

```
aws autoscaling describe-auto-scaling-instances
```

   以下为响应示例。

```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-05b4f7d5be44822a6",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
       ...
    ]
}
```

2. 将实例移动到 Standby 状态使用以下 enter-standby 命令。--should-decrement-desired-capacity 选项将减少所需容量以使 Auto Scaling 组不再启动替代实例。

```
aws autoscaling enter-standby --instance-ids i-05b4f7d5be44822a6 \
  --auto-scaling-group-name my-asg --should-decrement-desired-capacity
```

以下为响应示例。

```
{
    "Activities": [
        {
            "Description": "Moving EC2 instance to Standby: i-05b4f7d5be44822a6",
            "AutoScalingGroupName": "my-asg",
            "ActivityId": "3b1839fe-24b0-40d9-80ae-bcd883c2be32",
            "Details": "{\"Availability Zone\":\"us-west-2a\"}",
            "StartTime": "2014-12-15T21:31:26.150Z",
            "Progress": 50,
            "Cause": "At 2014-12-15T21:31:26Z instance i-05b4f7d5be44822a6 was moved to
 standby
              in response to a user request, shrinking the capacity from 4 to 3.",
            "StatusCode": "InProgress"
        }
    ]
}
```

3. （可选）使用 describe-auto-scaling-instances 命令验证实例是否处于 Standby。

```
aws autoscaling describe-auto-scaling-instances --instance-ids i-05b4f7d5be44822a6
```

以下为响应示例。注意，实例状态此时为 Standby。

```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-05b4f7d5be44822a6",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "Standby"
        },
       ...
    ]
}
```

4. 您可以根据需要更新实例或排查实例的问题。当您完成后，请继续下一步以将实例恢复运行。
5. 使用以下 exit-standby 命令使实例恢复运行：

```
aws autoscaling exit-standby --instance-ids i-05b4f7d5be44822a6 --auto-scaling-group-
name my-asg
```

以下为响应示例。

```
{
    "Activities": [
        {
            "Description": "Moving EC2 instance out of Standby: i-05b4f7d5be44822a6",
            "AutoScalingGroupName": "my-asg",
            "ActivityId": "db12b166-cdcc-4c54-8aac-08c5935f8389",
            "Details": "{\"Availability Zone\":\"us-west-2a\"}",
            "StartTime": "2014-12-15T21:46:14.678Z",
```

```
            "Progress": 30,
            "Cause": "At 2014-12-15T21:46:14Z instance i-05b4f7d5be44822a6 was moved
 out of standby in
               response to a user request, increasing the capacity from 3 to 4.",
            "StatusCode": "PreInService"
        }
    ]
}
```

6. （可选）使用以下 `describe-auto-scaling-instances` 命令验证实例是否已恢复运行。

```
aws autoscaling describe-auto-scaling-instances --instance-ids i-05b4f7d5be44822a6
```

以下为响应示例。请注意，实例状态为 `InService`。

```
{
    "AutoScalingInstances": [
        {
            "ProtectedFromScaleIn": false,
            "AvailabilityZone": "us-west-2a",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-050555ad16a3f9c7f"
            },
            "InstanceId": "i-05b4f7d5be44822a6",
            "AutoScalingGroupName": "my-asg",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService"
        },
       ...
    ]
}
```

暂停和恢复 Auto Scaling 组的进程

本主题介绍如何暂停然后恢复您的 Auto Scaling 组的一个或多个扩展进程。例如，这样操作可能会便于您调查导致进程失败的配置问题，或者防止 Amazon EC2 Auto Scaling 在您更改自动扩缩组时将实例标记为运行不正常并予以替换。

目录

- 进程的类型 (p. 236)
- 注意事项 (p. 237)
- 暂停和恢复扩展进程（控制台） (p. 239)
- 暂停和恢复进程 (Amazon CLI) (p. 239)

> **Note**
>
> 除在启动时暂停外，Amazon EC2 Auto Scaling 也可以暂停多次启动实例失败的 Auto Scaling 组。这称为管理暂停。管理暂停最常用于符合以下条件的 Auto Scaling 组：连续尝试启动实例的时间超过 24 小时，但是未成功启动任何实例。您可以恢复由于管理原因被 Amazon EC2 Auto Scaling 暂停的进程。

进程的类型

暂停-恢复功能支持以下进程：

- Launch – 在自动扩缩组横向扩展时，或者在 Amazon EC2 Auto Scaling 因其他原因（例如将实例添加到暖池时）选择启动实例时，将实例添加到该组中。
- Terminate – 在自动扩缩组横向缩减时，或者在 Amazon EC2 Auto Scaling 因其他原因（例如，实例因超过其最长生命周期或未通过运行状况检查而被终止时）选择终止实例时，将实例从该组中移除。
- AddToLoadBalancer—在实例启动时，将其添加到附上的负载均衡器目标组或者经典负载均衡器。有关更多信息，请参阅使用 Elastic Load Balancing 跨 Auto Scaling 组中的实例分配流量。 (p. 272)。
- AlarmNotification – 接受来自与动态扩缩策略关联的 CloudWatch 警报的通知。有关更多信息，请参阅Amazon EC2 Auto Scaling 的动态扩缩 (p. 132)。
- AZRebalance – 在该组失去均衡时（例如，之前不可用的可用区恢复到正常状态时），在指定的所有可用区之间均匀调整该组中的 EC2 实例数量。有关更多信息，请参阅再平衡活动 (p. 6)。
- HealthCheck - 检查实例的运行状况，如果 Amazon EC2 或 Elastic Load Balancing 通知 Amazon EC2 Auto Scaling 实例运行状况不佳，则将对应实例标记为运行状况不佳。此流程可覆盖您手动设置的实例运行状况状态。有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。
- InstanceRefresh-使用实例刷新功能终止和替换实例。有关更多信息，请参阅 基于实例刷新替换 Auto Scaling 实例 (p. 95)。
- ReplaceUnhealthy — 终止被标记为运行状况不佳的实例，然后创建新实例以替换它们。有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。
- ScheduledActions-执行您创建的计划扩展操作，或您在创建 Amazon Auto Scaling 扩展计划并启用预测扩展时创建的计划扩展操作。有关更多信息，请参阅Amazon EC2 Auto Scaling 的计划扩缩 (p. 186)。

注意事项

暂停进程之前，请注意以下事项：

- 您可以暂停和恢复单个流程或所有流程。
- 暂停进程将影响自动扩缩组中的所有实例。例如，您可以暂停 HealthCheck 和 ReplaceUnhealthy 进程以重启实例，但不让 Amazon EC2 Auto Scaling 根据运行状况检查的结果终止实例。如果需要 Amazon EC2 Auto Scaling 对剩余实例执行运行状况检查，则使用备用功能而不是暂停-恢复功能。有关更多信息，请参阅临时从 Auto Scaling 组中移除实例 (p. 232)。
- 通过暂停 AlarmNotification，您可以暂时停止该组的目标跟踪、步进和简单扩缩策略，而不删除扩缩策略或其关联的 CloudWatch 警报。要暂时停止单个扩缩策略，请参阅禁用 Auto Scaling 组的扩缩策略 (p. 161)。
- 如果您暂停了 Launch 和 Terminate 进程或者 AZRebalance，并且随后对自动扩缩组进行了更改（例如，分离实例或更改指定的可用区），则您的组可能会在可用区之间失去均衡。如果出现这种情况，在您恢复暂停的流程后，Amazon EC2 Auto Scaling 将逐步在可用区之间均匀地重新分配实例。
- 暂停 Terminate 过程不会以 delete-auto-scaling-group 命令通过强制删除选项阻止实例成功终止。
- 如果 RemoveFromLoadBalancerLowPriority 过程出现在使用 Amazon CLI 或开发工具包描述自动扩缩组的调用中，则应忽略该过程。此过程已过时，保留此过程仅是为了向后兼容。

了解暂停进程如何影响其他进程

下文说明暂停各个扩展类型时会发生什么。

场景 1：暂停 Launch

- AlarmNotification 仍处于活动状态，但是您的自动扩缩组无法为超限警报启动横向扩展活动。
- ScheduledActions 处于活动状态，但是您的自动扩缩组无法为出现的任何计划操作启动横向扩展活动。
- AZRebalance 停止对组进行重新平衡。

- ReplaceUnhealthy 继续终止运行不正常的实例，但不启动替换实例。Launch 过程恢复后，Amazon EC2 Auto Scaling 会立即替换它在 Launch 被暂停期间终止的所有实例。
- InstanceRefresh 不会替换实例。

场景 2：暂停 Terminate

- AlarmNotification 仍处于活动状态，但是您的自动扩缩组无法为超限警报启动横向缩减活动。
- ScheduledActions 处于活动状态，但是您的自动扩缩组无法为出现的任何计划操作启动横向缩减活动。
- AZRebalance 仍处于活动状态，但不能正常运行。它可以启动新实例而不终止旧实例。这可能导致您的 Auto Scaling 组增加到比最大大小超出百分之十，因为在重新平衡活动期间允许短时间内发生这种情况。您的 Auto Scaling 组可以保持超出其最大大小，直到您恢复 Terminate 进程。
- ReplaceUnhealthy 处于非活动状态但未 HealthCheck。当 Terminate 恢复后，ReplaceUnhealthy 进程将立即开始运行。如果任何实例在 Terminate 暂停期间被标记为运行状况不佳，将立即替换它们。
- InstanceRefresh 不会替换实例。

场景 3：暂停 AddToLoadBalancer

- Amazon EC2 Auto Scaling 会启动实例，但不会将实例添加到负载均衡器目标组或经典负载均衡器。在您恢复 AddToLoadBalancer 进程后，该进程也会在启动实例时将其添加到负载均衡器。不过，它不会添加在此流程暂停时启动的实例。您必须手动注册这些实例。

场景 4：暂停 AlarmNotification

- 当超出 CloudWatch 警报阈值时，Amazon EC2 Auto Scaling 不会调用扩缩策略。当您恢复 AlarmNotification 时，Amazon EC2 Auto Scaling 考虑具有当前违反的警报阈值的策略。

场景 5：暂停 AZRebalance

- Amazon EC2 Auto Scaling 不会在某些事件发生后尝试重新分配实例。如果发生扩展或缩减事件，扩展进程仍会尝试均衡可用区。例如，在扩展期间，它会在实例最少的可用区中启动实例。如果组变得不平衡，同时 AZRebalance 处于暂停状态并且您恢复它，Amazon EC2 Auto Scaling 会尝试重新平衡组。它先调用 Launch，然后调用 Terminate。

场景 6：暂停 HealthCheck

- Amazon EC2 Auto Scaling 停止在 EC2 和 Elastic Load Balancing 运行状况检查后将实例标记为运行状况不佳。您的自定义运行状况检查会继续正常运行。当您暂停 HealthCheck 后，在需要时可以手动设置组中实例的运行状况，并由 ReplaceUnhealthy 替换它们。

场景 7：暂停 InstanceRefresh

- Amazon EC2 Auto Scaling 停止因实例刷新而替换实例。如果正在刷新实例，则会暂停操作而不将其取消。

场景 8：暂停 ReplaceUnhealthy

- Amazon EC2 Auto Scaling 停止替换标记为运行状况不佳的实例。未通过 EC2 或 Elastic Load Balancing 运行状况检查的实例仍会被标记为运行状况不佳。一旦您恢复 ReplaceUnhealthy 进程，Amazon EC2 Auto Scaling 会替换在此进程暂停时标记为运行状况不佳的实例。ReplaceUnhealthy 进程会首先调用 Terminate，然后调用 Launch。

场景 9：暂停 ScheduledActions

- Amazon EC2 Auto Scaling 不会运行计划在暂停期间运行的计划操作。当您恢复 ScheduledActions 后，Amazon EC2 Auto Scaling 仅考虑其计划时间尚未过期的计划操作。

其它注意事项

此外，在暂停 Launch 或 Terminate 时，以下功能可能无法正常运行：

- Maximum instance lifetime（最大实例生命周期）– 暂停 Launch 或者 Terminate 时，则最大实例生命周期功能无法替换任何实例。
- 竞价型实例中断—如果 Terminate 已暂停而您的 Auto Scaling 组包含竞价型实例，它们在竞价容量不再可用的情况下仍会终止。暂停 Launch 时，Amazon EC2 Auto Scaling 无法从其他竞价型实例池启动替换实例，也无法在同一竞价型实例池再次可用时启动替换实例。
- Capacity Rebalancing（容量再平衡）– 如果暂停了 Terminate 并且您使用容量再平衡以编程方式处理竞价型实例中断，则当竞价型容量不再可用时，Amazon EC2 竞价型服务仍可以终止实例。如果暂停 Launch，Amazon EC2 Auto Scaling 无法从其他竞价型实例池启动替换实例，也无法在同一竞价型实例池再次可用时启动替换实例。
- Attaching and detaching instances（附加和分离实例）– 暂停 Launch 和 Terminate 时，您可以分离附加到自动扩缩组的实例，但在暂停 Launch 期间，您无法将新实例附加到该组。
- Standby instances（备用实例）– 暂停 Launch 和 Terminate 时，您可以将实例置于 Standby 状态，但在暂停 Launch 期间，您无法将处于 Standby 状态的实例重设为服务状态。

暂停和恢复扩展进程（控制台）

使用以下过程暂停进程。

暂停一个流程

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details（详细信息）选项卡上，选择 Advanced configurations（高级配置）、Edit（编辑）。
4. 对于 Suspended processes（暂停的进程），选择要暂停的进程。
5. 选择 Update（更新）。

准备就绪后，请使用以下过程恢复暂停的进程。

恢复进程

1. 在 Details（详细信息）选项卡上，选择 Advanced configurations（高级配置）、Edit（编辑）。
2. 对于 Suspended processes（已暂停的进程），移除该已暂停的进程。
3. 选择 Update（更新）。

暂停和恢复进程 (Amazon CLI)

使用下面的 suspend-processes 命令暂停单个进程。

```
aws autoscaling suspend-processes --auto-scaling-group-name my-asg --scaling-
processes HealthCheck ReplaceUnhealthy
```

要暂停所有进程，请忽略 `--scaling-processes` 选项，如下所示。

```
aws autoscaling suspend-processes --auto-scaling-group-name my-asg
```

准备就绪可以恢复已暂停的进程时，请使用下面的 resume-processes 命令。

```
aws autoscaling resume-processes --auto-scaling-group-name my-asg --scaling-
processes HealthCheck
```

要恢复所有已暂停的进程，请忽略 `--scaling-processes` 选项，如下所示。

```
aws autoscaling resume-processes --auto-scaling-group-name my-asg
```

监控 Auto Scaling 实例和组

监控是保持 Amazon EC2 Auto Scaling 和您的 Amazon Web Services Cloud 解决方案的可靠性、可用性和性能的重要方面。Amazon 提供了以下一些监控工具来监控 Amazon EC2 Auto Scaling、在出现错误时进行报告并适时自动采取措施。

运行状况检查

Amazon EC2 Auto Scaling 会对 Auto Scaling 组中的实例定期执行运行状况检查。如果某个实例未通过其运行状况检查，则该实例将被标记为运行状况不佳，并且在 Amazon EC2 Auto Scaling 启动新实例以替换实例时终止。有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。

Amazon Health Dashboard

Amazon Health Dashboard 会显示相关信息，并提供因 Amazon 资源的运行状况变化触发的通知。信息会以两种方式显示：在显示按类别组织的最近和未来事件的控制面板上，以及在显示过去 90 天内所有事件的完整事件日志中。有关更多信息，请参阅Amazon EC2 Auto Scaling 的 Amazon Health Dashboard 通知 (p. 247)。

CloudWatch alarms（CloudWatch 告警）

CloudWatch 可以自动监控 Amazon 资源的特定指标，从而帮助您检测不正常的应用程序行为。您可以配置 CloudWatch 警报并设置 Amazon SNS 通知，以在指标值不符合预期或者检测到特定异常时发送电子邮件。例如，您可以在网络活动突然高于或低于指标的预期值时收到通知。有关更多信息，请参阅监控 Auto Scaling 组和实例的 CloudWatch 指标 (p. 248)。

CloudWatch dashboards（CloudWatch 控制面板）

CloudWatch 控制面板是 CloudWatch 控制台中可自定义的主页。您可以使用这些页面在单个视图中监控您的资源，甚至包括分布在不同区域的资源。您可以使用 CloudWatch 控制面板创建 Amazon 资源的指标和警报的自定义视图。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

CloudTrail 日志

Amazon CloudTrail 让您能够跟踪由或代表您的 Amazon Web Services 账户对 Amazon EC2 Auto Scaling API 发起的调用。CloudTrail 将信息存储在您指定的 Amazon S3 存储桶中的日志文件中。您可以使用这些日志文件监控 Auto Scaling 组的活动。日志包括发出的请求、请求来自的源 IP 地址、发出请求的用户、发出请求的时间，等等。有关更多信息，请参阅使用 Amazon CloudTrail 记录 Amazon EC2 Auto Scaling API 调用 (p. 256)。

CloudWatch Logs

CloudWatch Logs 使您能够监控、存储和访问来自 Amazon EC2 实例、CloudTrail 和其他来源的日志文件。CloudWatch Logs 可以监控日志文件中的信息，并在达到特定阈值时通知您。您还可以在高持久性存储中检索您的日志数据。有关更多信息，请参阅 Amazon CloudWatch Logs 用户指南。

Amazon Simple Notification Service 通知

当 Amazon EC2 Auto Scaling 启动或终止实例时，您可以配置 Auto Scaling 组以发送 Amazon SNS 通知。有关更多信息，请参阅在 Auto Scaling 组扩展时获取 Amazon SNS 通知 (p. 258)。

Auto Scaling 实例的运行状况检查

Auto Scaling 实例的运行状态说明该实例运行正常或不正常。Auto Scaling 组中所有实例在开始启动时都处于良好运行状态。除非 Amazon EC2 Auto Scaling 收到实例运行状况不佳的通知，否则将认为它们运行状况

良好。此通知可能来自 Amazon EC2、Elastic Load Balancing 或自定义运行状况检查等来源。Amazon EC2 Auto Scaling 检测到任何运行不正常的实例时，它会将其终止并启动一个新实例。

目录

- 运行状况检查类型 (p. 242)
- Amazon EC2 运行状况检查 (p. 242)
- Elastic Load Balancing 运行状况检查 (p. 243)
- 自定义运行状况检测 (p. 244)
- 替换运行不正常的实例 (p. 244)
- Amazon EC2 Auto Scaling 如何尽可能减少停机时间 (p. 245)
- 运行状况检查注意事项 (p. 245)
- 其他信息 (p. 246)
- 设置自动扩缩组的运行状况检查宽限期 (p. 246)

运行状况检查类型

Amazon EC2 Auto Scaling 可以使用下列一项或多项运行状况检查来确定实例的运行状态：

运行状况检查类型	检查内容
Amazon EC2 状态检查和计划事件	• 检查实例是否正在运行 • 检查可能会影响实例的潜在硬件或软件问题 这是自动扩缩组的默认运行状况检查类型。
Elastic Load Balancing 运行状况检查	• 检查负载均衡器是否将实例报告为运行正常，以确认实例是否可用于处理请求 要运行此运行状况检查类型，您必须为自动扩缩组启用此运行状况检查。
自定义运行状况检查	• 根据自定义运行状况检查的结果，检查可能表明实例存在运行状况问题的其他问题。

Amazon EC2 运行状况检查

实例启动后，它会附加到 Auto Scaling 组并进入 `InService` 状态。要详细了解 Auto Scaling 组中实例的不同生命周期状态，请参阅 Amazon EC2 Auto Scaling 实例生命周期 (p. 7)。

Amazon EC2 Auto Scaling 通过定期检查自动扩缩组中所有实例的运行状态，以确保所有实例都在运行并且运行正常。

状态检查

Amazon EC2 Auto Scaling 使用 Amazon EC2 实例状态检查和系统状态检查的结果来确定实例的运行状况。如果实例状态是除 `running` 以外的其他 Amazon EC2 状态，或者状态检查的状态变为 `impaired`，则 Amazon EC2 Auto Scaling 会认为实例运行不正常并予以替换。这包括当实例为任何以下状态时：

- `stopping`

- stopped
- shutting-down
- terminated

Amazon EC2 状态检查不需要任何特殊配置，并且始终处于启用状态。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的状态检查类型。

> **Important**
>
> Amazon EC2 Auto Scaling 会允许这些状态检查偶尔失败，而不执行任何操作。当状态检查失败时，Amazon EC2 Auto Scaling 将等待几分钟，以便 Amazon 修复此问题。它不会在状态检查的状态变为 impaired 时立即将实例标记为运行不正常。
> 但如果 Amazon EC2 Auto Scaling 检测到实例不再处于 running 状态，这种情况将被视为立即失败。在这种情况下，它会立即将实例标记为运行不正常并予以替换。

计划的事件

Amazon EC2 偶尔可以将实例上的事件安排在特定时间戳之后运行。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的实例的计划事件。

如果其中一个实例受到计划事件的影响，则 Amazon EC2 Auto Scaling 会认为实例运行不正常并将其替换。在到达时间戳中指定的日期和时间之前，实例不会开始终止。

Elastic Load Balancing 运行状况检查

当您为自动扩缩组启用了 Elastic Load Balancing 运行状况检查时，Amazon EC2 Auto Scaling 还使用这些运行状况检查的结果来确定实例的运行状态。

您必须首先完成以下操作，然后才能为自动扩缩组启用 Elastic Load Balancing 运行状况检查：

- 设置一个 Elastic Load Balancing 负载均衡器并为其配置运行状况检查，以便该负载均衡器用来确定实例是否运行正常。
- 将负载均衡器附加到自动扩缩组

完成前提操作后，将会出现以下情况：

- Amazon EC2 Auto Scaling 会将自动扩缩组中的实例注册到该负载均衡器。
- 实例完成注册后，它会进入 InService 状态并可与该负载均衡器一起使用。

预设情况下，Amazon EC2 Auto Scaling 会忽略 Elastic Load Balancing 运行状况检查的结果。但是，您可以为自动扩缩组启用这些运行状况检查。完成此操作后，当 Elastic Load Balancing 报告某个注册的实例为 unhealthy 时，Amazon EC2 Auto Scaling 会在下一次定期运行状况检查中将实例标记为 unhealthy 并予以替换。

如果为负载均衡器启用了连接耗尽（注销延迟），则在终止运行不正常的实例之前，Amazon EC2 Auto Scaling 会等待进行中的请求完成或等待最大超时过期。

要了解详情，包括如何为自动扩缩组启用 Elastic Load Balancing 运行状况检查，请参阅向 Auto Scaling 组添加 Elastic Load Balancing 运行状况检查 (p. 276)。

> **Note**
>
> 当您为某个组启用 Elastic Load Balancing 运行状况检查时，Amazon EC2 Auto Scaling 可以终止和替换报告为运行不正常的实例，但仅在负载均衡器处于 InService 状态后才会执行此操作。有关更多信息，请参阅了解负载均衡器的附加状态 (p. 275)。

自定义运行状况检测

您还可能会需要在自动扩缩组中的实例上运行自定义运行状况检测任务，如果任务失败，则将实例的运行状况设为运行不正常。从而通过组合使用自定义运行状况检查、Amazon EC2 状态检查和 Elastic Load Balancing 运行状况检查（如果启用）来扩展运行状况检查的范围。

您可以使用 Amazon CLI 或 SDK 将实例运行状况信息直接发送到 Amazon EC2 Auto Scaling。以下示例演示如何使用 Amazon CLI 配置实例的运行状况，然后验证实例的运行状况。

使用以下 set-instance-health 命令，将指定实例的运行状态设置为 Unhealthy。

```
aws autoscaling set-instance-health --instance-id i-1234567890abcdef0 --health-status
 Unhealthy
```

默认情况下，此命令会执行运行状况检查宽限期。不过，您可以通过包括 --no-should-respect-grace-period 选项来覆盖此行为，不执行此宽限期。

使用以下 describe-auto-scaling-groups 命令验证实例状态为 Unhealthy。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg
```

下面是一个示例响应，说明实例的运行状况为 Unhealthy 并且实例正在终止。

```
{
    "AutoScalingGroups": [
        {
            ....
            "Instances": [
                {
                    "ProtectedFromScaleIn": false,
                    "AvailabilityZone": "us-west-2a",
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-1234567890abcdef0"
                    },
                    "InstanceId": "i-1234567890abcdef0",
                    "HealthStatus": "Unhealthy",
                    "LifecycleState": "Terminating"
                },
                ...
            ]
        }
    ]
}
```

替换运行不正常的实例

如果 Amazon EC2 Auto Scaling 确定某个 InService 实例运行不正常，则会将其终止并启动一个新的替换实例。新实例使用 Auto Scaling 组的当前设置及其关联的启动模板或启动配置启动。

Amazon EC2 Auto Scaling 将创建用于终止运行状况不佳的实例的新扩展活动，然后终止该实例。在实例终止期间，另一个扩缩活动将会启动一个新实例。

查看运行状况检查失败原因（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。

2. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在活动选项卡的活动历史记录下，状态列显示您的 Auto Scaling 组是否已成功启动或终止实例。

   如果它终止了任何运行状况不佳的实例，原因列显示终止的日期和时间以及运行状况检查失败的原因。例如，`At 2022-05-14T20:11:53Z an instance was taken out of service in response to an ELB system health check failure`。

Amazon EC2 Auto Scaling 如何尽可能减少停机时间

因运行状况检查的结果而替换实例时，需要首先终止实例，这可能导致在新实例启动之前无法接受新请求。

如果 Amazon EC2 Auto Scaling 确定任何实例不再运行或已由 set-instance-health 命令标记为运行不正常，则会立即将其替换。但如果发现其他实例运行不正常，Amazon EC2 Auto Scaling 将使用以下故障恢复方法。这种方法可尽可能减少可能因临时问题或运行状况检查配置错误导致的任何停机时间。

- 如果正在进行某个扩缩活动并且您的自动扩缩组低于所需容量 10% 或以上，则 Amazon EC2 Auto Scaling 会首先等待进行中的扩缩活动完成，然后再替换运行不正常的实例。
- 在进行横向扩展时，Amazon EC2 Auto Scaling 将首先等待实例通过初始运行状况检查。它还会等待默认实例预热完成，以确保新实例准备就绪。
- 在实例完成预热并且该组升至所需容量的 90% 以上后，Amazon EC2 Auto Scaling 将替换运行不正常的实例，如下所示：
  - Amazon EC2 Auto Scaling 一次只替换该组所需容量的 10%。它将在所有运行不正常的实例都被替换之前持续这样操作。
  - 在替换实例时，它会等待新实例通过初始运行状况检查。它还会等待默认的实例预热期结束，然后再继续操作。

  Note

  如果某个自动扩缩组的容量非常小，以至于容量的 10% 小于 1，则 Amazon EC2 Auto Scaling 将改为一次替换一个运行不正常的实例。这可能会导致该组出现短暂停机。
  此外，如果 Elastic Load Balancing 运行状况检查报告自动扩缩组中的所有实例都不正常，并且负载均衡器位于 `InService` 状态，则 Amazon EC2 Auto Scaling 可能会减少一次标记为运行不正常的实例数量。这可能会导致一次替换的实例数量少于在其他场景中执行的 10% 这一比例。从而您有时间来解决问题，无需 Amazon EC2 Auto Scaling 自动终止整个组。

运行状况检查注意事项

这一部分包含了有关 Amazon EC2 Auto Scaling 运行状况检查的注意事项。

- 如果您需要在正在终止的实例上或正在启动的实例上执行某些操作，则可以使用生命周期钩子。借助这些钩子，您可以在 Amazon EC2 Auto Scaling 启动或终止实例时执行自定义操作。有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。
- Amazon EC2 Auto Scaling 没有提供从运行状况检查中移除 Amazon EC2 状态检查和计划事件的方法。如果您不想替换实例，则建议暂停任何单个 Auto Scaling 组的 `ReplaceUnhealthy` 和 `HealthCheck` 进程。有关更多信息，请参阅暂停和恢复 Auto Scaling 组的进程 (p. 236)。
- 您可以尝试使用 set-instance-health 命令，从而手动将运行不正常实例的运行状态重新设置为正常。如果您收到错误消息，则可能是因为该实例已经开始终止。通常，使用 set-instance-health 命令将实例的运行状态重新设置为正常的方法仅在 `ReplaceUnhealthy` 进程或 `Terminate` 进程已暂停时有用。
- Amazon EC2 Auto Scaling 不会对处于 `Standby` 状态的实例执行运行状况检查。有关更多信息，请参阅临时从 Auto Scaling 组中移除实例 (p. 232)。

- 实例终止后，任何关联的弹性 IP 地址都会取消关联，并且不会自动与新实例关联。必须手动将弹性 IP 地址关联到新实例，或者使用基于生命周期挂钩的解决方案自动完成关联。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的弹性 IP 地址。
- 同样，实例终止后，其挂载的 EBS 卷也将被分离（或删除，具体取决于卷的 `DeleteOnTermination` 属性）。必须手动将这些 EBS 卷挂载到新实例，或者使用基于生命周期挂钩的解决方案自行完成挂载。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的将 Amazon EBS 卷挂载到实例。

其他信息

有关排查运行状况检查问题的详细信息，请参阅 Amazon EC2 Auto Scaling 问题排查：运行状况检查 (p. 351)。如果运行状况检查失败，请查看本主题以了解故障排查步骤。本主题将帮助您弄明白 Auto Scaling 组中发生了什么故障，并提供相关的修复建议。

Amazon EC2 Auto Scaling 还会使用 Amazon EC2、Amazon EBS 或自定义运行状况检查来监控您在暖池中启动的实例的运行状况。有关更多信息，请参阅查看运行状况检查状态以及运行状况检查失败的原因 (p. 218)。

设置自动扩缩组的运行状况检查宽限期

如果 Amazon EC2 Auto Scaling 运行状况检查确定某个 `InService` 实例运行不正常，则会将其终止并启动一个新的替换实例。运行状况检查宽限期指定了因新实例运行状况不正常而将其终止之前继续运行的最短时间（以秒为单位）。

一个示例用例可能是要求 Amazon EC2 Auto Scaling 避免在 Elastic Load Balancing 运行状况检查失败且原因是实例仍在初始化时采取行动。Elastic Load Balancing 运行状况检查并行运行，从实例向负载均衡器注册时开始。通过设置宽限期，可以防止 Amazon EC2 Auto Scaling 在您新启动的实例进入 `InService` 状态后，但没有立即通过这些运行状况检查时，不必要地将这些实例标记为运行不正常并予以终止。

在控制台中，创建自动扩缩组时的运行状况检查宽限期默认为 300 秒。使用 Amazon CLI 或 SDK 创建自动扩缩组时，默认值为 0 秒。

如果将该值设置过高，会降低 Amazon EC2 Auto Scaling 运行状况检查的效果。如果您为实例启动使用了生命周期挂钩，则可以将运行状况检查宽限期设置为 0。借助生命周期挂钩，Amazon EC2 Auto Scaling 提供了一种方法，可确保实例始终在完成初始化之后才会进入 `InService` 状态。有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

宽限期适用于以下实例：

- 新启动的实例
- 处于待机状态后重新投入运行的实例
- 您手动附加到组的实例

> **Important**
>
> 在运行状况检查宽限期内，如果 Amazon EC2 Auto Scaling 检测到某个实例不再处于 Amazon EC2 `running` 状态，则会立即将该实例标记为运行不正常并予以替换。例如，假设您停止了自动扩缩组中的某个实例，则该实例会被标记为运行不正常并被替换。

设置组的运行状况检查宽限期

您可以为新的和现有的自动扩缩组设置运行状况检查宽限期。

Console

修改新组的运行状况检查宽限期（控制台）

创建自动扩缩组时，在 Configure advanced options（配置高级选项）页面上，对于 Health checks（运行状况检查）的 Health check grace period（运行状况检查宽限期），输入相应的时间（单位为秒）。这是 Amazon EC2 Auto Scaling 在实例进入 `InService` 状态后需要等待完成实例运行状况检查的时间。

Amazon CLI

修改新组的运行状况检查宽限期（Amazon CLI）

将 `--health-check-grace-period` 选项添加到 create-auto-scaling-group 命令。以下示例将为一个名为 *my-asg* 的新自动扩缩组配置运行状况检查宽限期，其值为 *60* 秒。

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --health-
check-grace-period 60 ...
```

Console

修改现有组的运行状况检查宽限期（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 在屏幕顶部的导航栏中，选择您在其中创建了自动扩缩组的 Amazon Web Services 区域。
3. 选中 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
4. 在 Details (详细信息) 选项卡上，选择 Health checks (运行状况检查)、Edit (编辑)。
5. 在 Health check grace period（运行状况检查宽限期）下，输入时间长短，单位为秒。这是 Amazon EC2 Auto Scaling 在实例进入 `InService` 状态后需要等待完成实例运行状况检查的时间。
6. 选择 Update（更新）。

Amazon CLI

修改现有组的运行状况检查宽限期（Amazon CLI）

将 `--health-check-grace-period` 选项添加到 update-auto-scaling-group 命令。以下示例将为一个名为 *my-asg* 的现有自动扩缩组配置运行状况检查宽限期，其值为 *120* 秒。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --health-
check-grace-period 120
```

Note

我们强烈建议同时为您的自动扩缩组设置默认实例预热时间。有关更多信息，请参阅为 Auto Scaling 组设置原定设置实例预热 (p. 150)。

Amazon EC2 Auto Scaling 的 Amazon Health Dashboard 通知

您的 Amazon Health Dashboard 支持来自 Amazon EC2 Auto Scaling 的通知。针对可能影响应用程序的资源性能或者可用性问题，这些通知可以让您注意相关情况并提供修正指导。当前只有特定于缺少安全组和启动模板的事件可用。

Amazon Health Dashboard 是 Amazon Health 服务的一部分。它不需要设置，您的账户中通过身份验证的任何用户都可以查看。有关更多信息，请参阅 Amazon Health Dashboard 入门。

如果您收到类似于下文的消息，则应将其视为警报并采取措施。

示例：由于缺少安全组，Auto Scaling 组未向外扩展

```
Hello,

At 2020-01-11 04:00 UTC, we detected an issue with your Auto Scaling group [ARN] in
Amazon Web Services ## 123456789012.

A security group associated with this Auto Scaling group cannot be found. Each time a
scale out operation is performed, it will be prevented until you make a change that
fixes the issue.

We recommend that you review and update your Auto Scaling group configuration to change
the launch template or launch configuration that depends on the unavailable security
group.

Sincerely,
Amazon Web Services
```

示例：由于缺少启动模板，Auto Scaling 组未向外扩展

```
Hello,

At 2021-05-11 04:00 UTC, we detected an issue with your Auto Scaling group [ARN] in
Amazon Web Services ## 123456789012.

The launch template associated with this Auto Scaling group cannot be found. Each time
a scale out operation is performed, it will be prevented until you make a change that
fixes the issue.

We recommend that you review and update your Auto Scaling group configuration and
specify an existing launch template to use.

Sincerely,
Amazon Web Services
```

监控 Auto Scaling 组和实例的 CloudWatch 指标

指标是 Amazon CloudWatch 中的基本概念。指标表示一个发布到 CloudWatch 并且按时间排序的数据点集。可将指标视为要监控的变量，而数据点代表该变量随时间变化的值。您可使用这些指标来验证您的系统是否按预期运行。

用于收集有关自动扩缩组信息的Amazon EC2 Auto Scaling 指标位于 `AWS/AutoScaling` 命名空间中。收集 Auto Scaling 实例的 CPU 和其他使用情况数据的 Amazon EC2 实例指标位于 `AWS/EC2` 命名空间中。

Amazon EC2 Auto Scaling 控制台会根据组指标和该组的聚合实例指标显示一系列的图表。根据您的需求，您可能更适合从 Amazon CloudWatch 而非 Amazon EC2 Auto Scaling 控制台访问您的自动扩缩组和实例数据。

有关更多信息，请参阅 Amazon CloudWatch 用户指南。

目录

- 可用指标和维度 (p. 249)
- 启用 Auto Scaling 组指标（控制台）(p. 252)

- 启用 Auto Scaling 组指标 (Amazon CLI) (p. 252)
- 配置 Auto Scaling 实例的监控 (p. 253)
- 在 Amazon EC2 Auto Scaling 控制台中查看监控图表 (p. 254)

可用指标和维度

Note

本部分列出了 AWS/AutoScaling 命名空间中不同类型的指标。
有关 AWS/EC2 命名空间中可用指标的信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的列出实例的可用 CloudWatch 指标。有关启用详细监控以在 1 分钟内向 CloudWatch 发送 EC2 使用数据，或向 CloudWatch 发送内存使用数据的信息，请参阅 配置 Auto Scaling 实例的监控 (p. 253)。

Amazon EC2 Auto Scaling 会在 AWS/AutoScaling 命名空间中发布以下指标。

Amazon EC2 Auto Scaling 会尽力每隔一分钟将样本数据发送到 CloudWatch。发生 CloudWatch 服务中断的极少数情况时，系统不会回填数据以填补组指标历史记录中的空缺。

目录

- Auto Scaling 组指标 (p. 249)
- Auto Scaling 组指标的维度 (p. 251)
- 预测性扩缩指标和维度 (p. 251)

Auto Scaling 组指标

启用组指标时，Amazon EC2 Auto Scaling 会向 CloudWatch 发送以下指标。这些指标以一分钟的粒度提供，无需额外付费，但您必须启用它们。借助这些指标，您可以更清楚地了解自动扩缩组的历史记录，例如组大小随时间推移的变化。

指标	描述
GroupMinSize	自动扩缩组的最小大小。 报告条件：如果启用了指标收集，则报告。
GroupMaxSize	自动扩缩组的最大大小。 报告条件：如果启用了指标收集，则报告。
GroupDesiredCapacity	Auto Scaling 组试图维护的实例数量。 报告条件：如果启用了指标收集，则报告。
GroupInServiceInstances	作为 Auto Scaling 组的一部分运行的实例数量。该指标不包括处于挂起或终止状态的实例。 报告条件：如果启用了指标收集，则报告。
GroupPendingInstances	处于挂起状态的实例数量。挂起的实例尚不可用。该指标不包括处于可用状态或终止状态的实例。 报告条件：如果启用了指标收集，则报告。
GroupStandbyInstances	处于 Standby 状态的实例数。处于此状态的实例仍在运行，但不能有效使用。

指标	描述
	报告条件：如果启用了指标收集，则报告。
GroupTerminatingInstances	正处于终止过程中的实例的数量。该指标不包括处于可用状态或挂起状态的实例。 报告条件：如果启用了指标收集，则报告。
GroupTotalInstances	Auto Scaling 组中的实例总数。该指标用于标识处于可用状态、挂起状态和终止状态的实例的数量。 报告条件：如果启用了指标收集，则报告。

除上表中的指标外，Amazon EC2 Auto Scaling 还会将组指标报告作为每个实例所代表容量单位数的聚合计数进行报告。如果未应用实例权重，则会填充以下指标，但等于上表中定义的指标。有关使用权重的更多信息，请参阅为 Amazon EC2 Auto Scaling 配置实例权重 (p. 67)和示例：使用基于属性的实例类型选择创建 Auto Scaling 组 (p. 72)。

指标	描述
GroupInServiceCapacity	作为 Auto Scaling 组的一部分运行的容量单位数量。 报告条件：如果启用了指标收集，则报告。
GroupPendingCapacity	待处理的容量单位数。 报告条件：如果启用了指标收集，则报告。
GroupStandbyCapacity	处于 Standby 状态的容量单位数。 报告条件：如果启用了指标收集，则报告。
GroupTerminatingCapacity	正处于终止过程中的容量单位的数量。 报告条件：如果启用了指标收集，则报告。
GroupTotalCapacity	Auto Scaling 组中的容量单位总数。 报告条件：如果启用了指标收集，则报告。

对于具有暖池的自动扩缩组，Amazon EC2 Auto Scaling 还会报告以下指标。有关更多信息，请参阅Amazon EC2 Auto Scaling 的暖池 (p. 212)。

指标	描述
WarmPoolMinSize	暖池的最小大小。 报告条件：如果启用了指标收集，则报告。
WarmPoolDesiredCapacity	Amazon EC2 Auto Scaling 尝试在暖池中维持的容量数。 这相当于 Auto Scaling 组的最大大小减去所需容量，或者等于 Auto Scaling 组的最大预热容量（如果设置）减去所需容量。 但是，如果暖池的最小大小等于或大于最大大小（或者是最大预热容量（如果设置））与 Auto Scaling 组的所需容量之间的差值，则暖池所需容量将等于 WarmPoolMinSize。

指标	描述
	报告条件：如果启用了指标收集，则报告。
WarmPoolPendingCapacity	暖池中待处理的容量数。该指标不包括处于运行、挂起或终止状态的实例。 报告条件：如果启用了指标收集，则报告。
WarmPoolTerminatingCapacity	暖池中处于终止过程的容量数。该指标不包括处于运行、已停止或挂起状态的实例。 报告条件：如果启用了指标收集，则报告。
WarmPoolWarmedCapacity	横向扩展期间可进入 Auto Scaling 组的容量数。该指标不包括处于挂起或终止状态的实例。 报告条件：如果启用了指标收集，则报告。
WarmPoolTotalCapacity	暖池的总容量，包括处于运行、已停止、挂起或终止状态的实例。 报告条件：如果启用了指标收集，则报告。
GroupAndWarmPoolDesiredCapacity	Auto Scaling 组和暖池结合起来的所需容量。 报告条件：如果启用了指标收集，则报告。
GroupAndWarmPoolTotalCapacity	Auto Scaling 组和暖池结合起来的总容量。这包括处于运行、已停止、挂起、终止或服务中状态的实例。 报告条件：如果启用了指标收集，则报告。

Auto Scaling 组指标的维度

您可以使用以下维度来优化上表中列出的指标。

维度	描述
AutoScalingGroupName	按 Auto Scaling 组名称筛选。

预测性扩缩指标和维度

AWS/AutoScaling 命名空间包括以下预测性扩缩指标。

提供精度为一小时的指标。

您可以通过将预测值与实际值进行比较来评估预测准确性。有关使用这些指标评估预测准确性的更多信息，请参阅使用 CloudWatch 监控预测性扩缩指标 (p. 174)。

指标	描述	维度
PredictiveScalingLoadForecast	应用程序预计将生成的负载量。 Average、Minimum 和 Maximum 统计数据非常有用，而 Sum 统计数据用处不大。 报告标准：在创建初始预测后报告。	AutoScalingGroupName, PolicyName, PairIndex

指标	描述	维度
PredictiveScalingCapacityForecast	满足应用程序需求所需的预期容量。这基于负载预测以及您希望 Auto Scaling 实例维持的目标利用率水平。 Average、Minimum 和 Maximum 统计数据非常有用，而 Sum 统计数据用处不大。 报告标准：在创建初始预测后报告。	AutoScalingGroupName, PolicyName
PredictiveScalingMetricPairCorrelation	扩展指标与负载指标的每个实例平均值之间的相关性。预测性扩展假设相关性很高。因此，如果您观察到该指标的值很低，最好不要使用指标对。 Average、Minimum 和 Maximum 统计数据非常有用，而 Sum 统计数据用处不大。 报告标准：在创建初始预测后报告。	AutoScalingGroupName, PolicyName, PairIndex

Note

PairIndex 维度会返回与 Amazon EC2 Auto Scaling 所分配负载扩缩指标对的索引相关的信息。目前唯一有效的值是 0。

启用 Auto Scaling 组指标（控制台）

启用 Auto Scaling 组指标时，Auto Scaling 组每分钟向 CloudWatch 发送采样数据。启用这些指标无需支付费用。

启用组指标

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在页面底部打开一个拆分窗格。
3. 在监控选项卡上，选择位于 Auto Scaling下页面顶部的 Auto Scaling 组指标集合、启用复选框。

禁用组指标

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选择您的 Auto Scaling 组。
3. 在监控选项卡上，清除 Auto Scaling 组指标集合、启用复选框。

启用 Auto Scaling 组指标 (Amazon CLI)

启用组指标

使用 enable-metrics-collection 命令启用一个或多个组指标。例如，以下命令可启用 GroupDesiredCapacity 指标。

```
aws autoscaling enable-metrics-collection --auto-scaling-group-name my-asg \
```

```
--metrics GroupDesiredCapacity --granularity "1Minute"
```

如果省略 `--metrics` 选项，则启用所有指标。

```
aws autoscaling enable-metrics-collection --auto-scaling-group-name my-asg \
--granularity "1Minute"
```

禁用组指标

请使用 disable-metrics-collection 命令。例如，以下命令可禁用所有 Auto Scaling 组指标。

```
aws autoscaling disable-metrics-collection --auto-scaling-group-name my-asg
```

配置 Auto Scaling 实例的监控

Amazon EC2 可从实例收集原始数据，并将数据处理为易读的近实时指标，以描述自动扩缩组的 CPU 和其他使用情况数据。对于间隔时间的配置，您可以选择按一分钟或五分钟的精度来监控这些指标。

每次启动实例时都会启用实例监控，可选择基本监控（五分钟精度）或详细监控（一分钟精度）。对于详细监控，将收取额外的费用。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 Amazon CloudWatch 定价和使用 CloudWatch 监控您的实例。

您首先需要创建一个启动模板或启动配置，以允许适合您的应用程序的监控类型，然后再创建自动扩缩组。如果您向组中添加扩缩策略，我们强烈建议您使用详细监控，以一分钟的精度获取 EC2 实例的指标数据，因为这可以更快地响应负载变化。

目录

- 启用详细监控（控制台） (p. 253)
- 启用详细监控 (Amazon CLI) (p. 253)
- 在基本监控和详细监控之间切换 (p. 254)
- 使用 CloudWatch 代理收集其他指标 (p. 254)

启用详细监控（控制台）

预设情况下，使用 Amazon Web Services Management Console 创建启动模板或启动配置时，将启用基本监控。

在启动模板中启用详细监控

使用 Amazon Web Services Management Console 创建启动模板时，请在高级详细信息部分中为禁用 CloudWatch 监控选择启用。否则，将启用基本监控。有关更多信息，请参阅配置启动模板的高级设置 (p. 24)。

在启动配置中启用详细监控

使用 Amazon Web Services Management Console 创建启动配置时，在其他配置部分中，选择在 CloudWatch 中启用 EC2 实例详细监控。否则，将启用基本监控。有关更多信息，请参阅创建启动配置 (p. 39)。

启用详细监控 (Amazon CLI)

预设情况下，使用 Amazon CLI 创建启动模板时，将启用基本监控。在使用 Amazon CLI 创建启动配置时，将默认启用详细监控。

在启动模板中启用详细监控

对于启动模板，请使用 create-launch-template 命令，并传递一个包含用于创建启动模板的信息的 JSON 文件。将监控属性设置为 `"Monitoring":{"Enabled":true}` 以启用详细监控，或设置为 `"Monitoring":{"Enabled":false}` 以启用基本监控。

在启动配置中启用详细监控

对于启动配置，请使用 create-launch-configuration 命令和 `--instance-monitoring` 选项。将此选项设置为 `true` 可启用详细监控，将此选项设置为 `false` 可启用基本监控。

```
--instance-monitoring Enabled=true
```

在基本监控和详细监控之间切换

要更改在新 EC2 实例上启用的监控类型，请更新启动模板或更新 Auto Scaling 组以使用新的启动模板或启动配置。现有实例将继续使用以前启用的监控类型。要更新所有实例，请终止这些实例，以便您的 Auto Scaling 组可以替换这些实例，或使用 monitor-instances 和 unmonitor-instances 逐一更新实例。

> **Note**
>
> 借助实例刷新和最长实例生命周期和功能，您还可以替换 Auto Scaling 组中的所有实例，以启动使用新设置的新实例。有关更多信息，请参阅替换 Auto Scaling 实例 (p. 94)。

在基本监控和详细监控之间切换时：

如果具有与 Auto Scaling 组关联的 CloudWatch 告警，请使用 put-metric-alarm 命令更新每个告警。使每个时段与监控类型匹配（基本监控为 300 秒，详细监控为 60 秒）。如果从详细监控更改为基本监控，但未更新警报以与 5 分钟时间段匹配，这些警报将继续每分钟检查一次统计数据。在每个 5 分钟时间段内，这些警报可能会在 4 分钟内找不到可用的数据。

使用 CloudWatch 代理收集其他指标

要收集可用内存和已使用内存等操作系统级别的指标，您必须安装 CloudWatch 代理。可能会产生额外的费用。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的 CloudWatch 代理收集的指标。

在 Amazon EC2 Auto Scaling 控制台中查看监控图表

在 Amazon EC2 控制台的 Amazon EC2 Auto Scaling 部分，您可以使用 CloudWatch 指标每分钟监控个别 Auto Scaling 组的进度。

您可以监控如下指标：

- Auto Scaling 指标 – 只有在启用的情况下才会开启这些 Auto Scaling 指标。有关更多信息，请参阅启用 Auto Scaling 组指标（控制台）(p. 252)。启用 Auto Scaling 指标后，监控图表显示以一分钟粒度发布的 Auto Scaling 指标数据。
- EC2 指标 – 如果启用了详细监控，监控图表将显示以一分钟精度发布的实例指标数据。有关更多信息，请参阅配置 Auto Scaling 实例的监控 (p. 253)。

使用 Amazon EC2 Auto Scaling 控制台查看监控图表

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中要查看其指标的 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling 组页面底部打开一个拆分窗格。
3. 选择监控选项卡。

Amazon EC2 Auto Scaling 显示 Auto Scaling 指标的监控图表。

4. 要查看组中所有实例的聚合数据监控图表，请选择 EC2。

图表操作

- 将鼠标悬停在数据点上，可以查看 UTC 中特定时间的数据弹出窗口。
- 要放大图表，请从图表右上角的菜单工具选择 Enlarge (放大) (三个垂直点) 。或者，选择图表顶部的最大化图标。
- 通过选择一个预定义的时间段值，调整图表中所显示数据的时间段。如果图表已放大，您可以选择 Custom (自定义) 来定义自己的时间段。
- 从菜单工具中选择 Refresh (刷新) 以更新图表中的数据。
- 将光标拖动到图表数据上以选择特定范围。然后，您可以选择工具菜单中的 Apply time range (应用时间范围)。
- 从菜单工具中选择 View logs (查看日志) 以查看 CloudWatch 控制台中关联的日志流 (如果有) 。
- 要在 CloudWatch 中查看图表，请从菜单工具中选择 View in metrics (在指标中查看)。这会将您引导至 CloudWatch 页面查看该图表。从中可以查看更多信息或访问历史信息，以更好地了解 Auto Scaling 组长时间内的变化情况。

适用于 Auto Scaling 组的图表指标

创建 Auto Scaling 组后，您可以打开 Amazon EC2 Auto Scaling 控制台并在 Monitoring (监控) 选项卡上查看该组的监控图表。

在 Auto Scaling 部分中，图表指标包括以下总体指标。这些指标提供的测量值可以指示潜在的问题，例如终止实例的数量或挂起实例的数量。

- Minimum Group Size (最小组大小) (基于 `GroupMinSize`)
- Maximum Group Size (最大组大小) (基于 `GroupMaxSize`)
- Desired Capacity (所需容量) (基于 `GroupDesiredCapacity`)
- In Service Instances (服务中实例) (基于 `GroupInServiceInstances`)
- Pending Instances (挂起实例) (基于 `GroupPendingInstances`)
- Standby Instances (备用实例) (基于 `GroupStandbyInstances`)
- Terminating Instances (终止实例) (基于 `GroupTerminatingInstances`)
- Total Instances (实例总数) (基于 `GroupTotalInstances`)

在 EC2 部分中，您可以根据 Amazon EC2 实例的关键性能指标找到以下图表指标。这些 EC2 指标是组中所有实例的指标聚合。

- CPU Utilization (CPU 利用率) (基于 `CPUUtilization`)
- Disk Reads (磁盘读取数) (基于 `DiskReadBytes`)
- Disk Read Operations (磁盘读取操作) (基于 `DiskReadOps`)
- Disk Writes (磁盘写入数) (基于 `DiskWriteBytes`)
- Disk Write Operations (磁盘写入操作) (基于 `DiskWriteOps`)
- Network In (网络输入) (基于 `NetworkIn`)
- Network Out (网络输出) (基于 `NetworkOut`)
- Status Check Failed (Any) (状态检查失败(任意)) (基于 `StatusCheckFailed`)
- Status Check Failed (Instance) (状态检查失败(实例)) (基于 `StatusCheckFailed_Instance`)
- Status Check Failed (System) (状态检查失败(系统)) (基于 `StatusCheckFailed_System`)

您可以使用下一组 Auto Scaling 图表指标以衡量特定组的特定功能。

对于实例具有定义每个实例对组所需容量贡献多少容量单位的权重的组，以下指标数据可用：

- In Service Capacity Units (服务中容量单位) (基于 `GroupInServiceCapacity`)
- Pending Capacity Units (挂起容量单位) (基于 `GroupPendingCapacity`)
- Standby Capacity Units (备用容量单位) (基于 `GroupStandbyCapacity`)
- Terminating Capacity Units (终止容量单位) (基于 `GroupTerminatingCapacity`)
- Total Capacity Units (总容量单位) (基于 `GroupTotalCapacity`)

如果组使用暖池 (p. 212)功能，则以下指标数据可用：

- Warm Pool Minimum Size (暖池最小大小) (基于 `WarmPoolMinSize`)
- Warm Pool Desired Capacity (暖池所需容量) (基于 `WarmPoolDesiredCapacity`)
- Warm Pool Pending Capacity Units (暖池挂起容量单位) (基于 `WarmPoolPendingCapacity`)
- Warm Pool Terminating Capacity Units (暖池终止容量单位) (基于 `WarmPoolTerminatingCapacity`)
- Warm Pool Warmed Capacity Units (暖池预热容量单位) (基于 `WarmPoolWarmedCapacity`)
- Warm Pool Total Capacity Units Launched (已启动的暖池总容量单位) (基于 `WarmPoolTotalCapacity`)
- Group and Warm Pool Desired Capacity (组和暖池所需容量) (基于 `GroupAndWarmPoolDesiredCapacity`)
- Group and Warm Pool Total Capacity Units Launched (已启动的组和暖池总容量单位 (基于 `GroupAndWarmPoolTotalCapacity`)

另请参阅

有关 EC2 图表指标数据源的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html中的列出实例的可用 CloudWatch 指标。

要监控各个实例的指标，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/graphs-in-the-aws-management-console.html中的实例的图表指标。

使用 Amazon CloudTrail 记录 Amazon EC2 Auto Scaling API 调用

Amazon EC2 Auto Scaling 与 Amazon CloudTrail 集成，后者是使用 Amazon EC2 Auto Scaling 提供用户、角色或服务所采取操作的记录的服务。CloudTrail 将 Amazon EC2 Auto Scaling 的所有 API 调用作为事件捕获。捕获的调用包括来自 Amazon EC2 Auto Scaling 控制台的调用以及对 Amazon EC2 Auto Scaling API 的代码调用。

如果您创建跟踪，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶，包括 Amazon EC2 Auto Scaling 的事件。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history (事件历史记录) 中查看最新事件。使用由 CloudTrail 收集的信息，您可以确定向 Amazon EC2 Auto Scaling 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

要了解有关 CloudTrail 的更多信息，请参阅《Amazon CloudTrail 用户指南》。

CloudTrail 中的 Amazon EC2 Auto Scaling 信息

创建账户时，将在 Amazon Web Services 账户上启用 CloudTrail。当 Amazon EC2 Auto Scaling 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 Amazon Web Services 事件一同保存在事件历史记录中。您可以在 Amazon Web Services 账户中查看、搜索和下载最新事件。有关更多信息，请参阅使用 CloudTrail 事件历史记录查看事件。

要持续记录 Amazon Web Services 账户中的事件（包括 Amazon EC2 Auto Scaling 的事件），请创建跟踪记录。通过跟踪记录，CloudTrail 可将日志文件传送至 Simple Storage Service（Amazon S3）存储桶。默认情况下，在控制台中创建跟踪时，此跟踪应用于所有区域。此跟踪记录来自 Amazon Web Services 分区中的所有区域的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 Amazon Web Services 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- 创建跟踪概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件

所有 Amazon EC2 Auto Scaling 操作都由 CloudTrail 记录，并记录在 Amazon EC2 Auto Scaling API 参考中。例如，对 CreateLaunchConfiguration、DescribeAutoScalingGroup 和 UpdateAutoScalingGroup 操作进行的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 Amazon Identity and Access Management（IAM）用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其它服务发出。

有关更多信息，请参阅 CloudTrail `userIdentity` 元素。

了解 Amazon EC2 Auto Scaling 日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 CreateLaunchConfiguration 操作。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-08-21T17:05:42Z"
            }
        }
    },
    "eventTime": "2018-08-21T17:07:49Z",
```

```
    "eventSource": "autoscaling.amazonaws.com",
    "eventName": "CreateLaunchConfiguration",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Coral/Jakarta",
    "requestParameters": {
        "ebsOptimized": false,
        "instanceMonitoring": {
            "enabled": false
        },
        "instanceType": "t2.micro",
        "keyName": "EC2-key-pair-oregon",
        "blockDeviceMappings": [
            {
                "deviceName": "/dev/xvda",
                "ebs": {
                    "deleteOnTermination": true,
                    "volumeSize": 8,
                    "snapshotId": "snap-01676e0a2c3c7de9e",
                    "volumeType": "gp2"
                }
            }
        ],
        "launchConfigurationName": "launch_configuration_1",
        "imageId": "ami-6cd6f714d79675a5",
        "securityGroups": [
            "sg-00c429965fd921483"
        ]
    },
    "responseElements": null,
    "requestID": "0737e2ea-fb2d-11e3-bfd8-99133058e7bb",
    "eventID": "3fcfb182-98f8-4744-bd45-b38835ab61cb",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```


在 Auto Scaling 组扩展时获取 Amazon SNS 通知

当 Amazon EC2 Auto Scaling 启动或终止 Auto Scaling 组中的 EC2 实例时，您可以收到通知。您可以使用 Amazon Simple Notification Service (Amazon SNS) 管理通知。

Amazon SNS 协调并管理传输，或将通知发送到订阅客户端或终端节点。Amazon SNS 提供了各种通知选项，包括以 HTTP 或 HTTPS POST、电子邮件（SMTP，采用纯文本或 JSON 格式）的形式传输通知，或将通知作为消息发布到 Amazon SQS 队列，从而使您能够以编程方式处理这些通知。有关更多信息，请参阅 Amazon Simple Notification Service 开发人员指南。

例如，如果将 Auto Scaling 组配置为使用 `autoscaling: EC2_INSTANCE_TERMINATE` 通知类型，并且您的 Auto Scaling 组终止了某个实例，则它会发送电子邮件通知。该电子邮件包含已终止实例的详细信息，如实例 ID 以及终止该实例的原因。

通知有助于设计事件驱动型应用程序。如果您使用通知来检查资源是否进入所需状态，则可以不再使用轮询，并且您不会遇到 `RequestLimitExceeded` 错误，这有时是由轮询引起的。

Amazon 提供了各种可用于发送通知的工具。或者，您可以使用 EventBridge 和 Amazon SNS 在您的 Auto Scaling 组启动或终止实例时发送通知。在 EventBridge 中，该规则描述向您通知的事件。在 Amazon SNS 中，该主题描述您接收的通知的类型。使用此选项，您可以决定某些事件是否应该触发 Lambda 函数。有关更多信息，请参阅使用 EventBridge 处理 Auto Scaling 事件 (p. 287)。

目录

- SNS 通知 (p. 259)

- 为 Amazon EC2 Auto Scaling 配置 Amazon SNS 通知 (p. 259)
  - 创建 Amazon SNS 主题 (p. 260)
  - 订阅 Amazon SNS 主题 (p. 260)
  - 确认您的 Amazon SNS 订阅 (p. 260)
  - 配置 Auto Scaling 组以发送通知 (p. 260)
  - 测试通知 (p. 261)
  - 删除通知配置 (p. 261)

SNS 通知

Amazon EC2 Auto Scaling 支持在发生以下事件时发送 Amazon SNS 通知。

事件	描述
autoscaling:EC2_INSTANCE_LAUNCH	实例启动成功
autoscaling:EC2_INSTANCE_LAUNCH_ERROR	实例启动失败
autoscaling:EC2_INSTANCE_TERMINATE	实例终止成功
autoscaling:EC2_INSTANCE_TERMINATE_ERROR	实例终止失败

消息包含以下信息：

- Event — 事件。
- AccountId — Amazon Web Services 账户 ID。
- AutoScalingGroupName — Auto Scaling 组的名称。
- AutoScalingGroupARN — Auto Scaling 组的 ARN。
- EC2InstanceId — EC2 实例的 ID。

例如：

```
Service: AWS Auto Scaling
Time: 2016-09-30T19:00:36.414Z
RequestId: 4e6156f4-a9e2-4bda-a7fd-33f2ae528958
Event: autoscaling:EC2_INSTANCE_LAUNCH
AccountId: 123456789012
AutoScalingGroupName: my-asg
AutoScalingGroupARN: arn:aws:autoscaling:region:123456789012:autoScalingGroup...
ActivityId: 4e6156f4-a9e2-4bda-a7fd-33f2ae528958
Description: Launching a new EC2 instance: i-0598c7d356eba48d7
Cause: At 2016-09-30T18:59:38Z a user request update of AutoScalingGroup constraints to ...
StartTime: 2016-09-30T19:00:04.445Z
EndTime: 2016-09-30T19:00:36.414Z
StatusCode: InProgress
StatusMessage:
Progress: 50
EC2InstanceId: i-0598c7d356eba48d7
Details: {"Subnet ID":"subnet-id","Availability Zone":"zone"}
```

为 Amazon EC2 Auto Scaling 配置 Amazon SNS 通知

要使用 Amazon SNS 发送电子邮件通知，必须先创建一个主题，然后用您的电子邮件地址订阅该主题。

创建 Amazon SNS 主题

SNS 主题是一个逻辑接入点，即 Auto Scaling 组用来发送通知的通信通道。您可通过为主题指定名称来创建主题。

您在创建主题名称时，该名称必须满足以下要求：

- 介于 1 到 256 个字符之间
- 包含大写和小写 ASCII 字母、数字、下划线或连字符

有关更多信息，请参阅 Amazon Simple Notification Service 开发人员指南中的创建 Amazon SNS 主题。

订阅 Amazon SNS 主题

要接收您的 Auto Scaling 组发送到该主题的通知，必须让一个终端节点订阅该主题。在此过程中，为端点指定要用来接收来自 Amazon EC2 Auto Scaling 的通知的电子邮件地址。

有关更多信息，请参阅 Amazon Simple Notification Service 开发人员指南中的订阅 Amazon SNS 主题。

确认您的 Amazon SNS 订阅

Amazon SNS 向在上一步骤中指定的电子邮件地址发送确认电子邮件。

确保打开并选择链接以确认订阅，然后再继续执行下一步。

您将收到来自 Amazon 的确认消息。Amazon SNS 现已配置为接收通知并以电子邮件形式将通知发送到指定的电子邮件地址。

配置 Auto Scaling 组以发送通知

您可以配置 Auto Scaling 组，以便在发生扩展事件（例如，启动实例或终止实例）时向 Amazon SNS 发送通知。Amazon SNS 向您指定的电子邮件地址发送通知，通知中包含有关实例的信息。

为 Auto Scaling 组配置 Amazon SNS 通知（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在页面底部打开一个拆分窗格，其中显示有关所选组的信息。
3. 在活动选项卡上，选择活动通知、创建通知。
4. 在 Create notifications 窗格上，执行以下操作：

   a. 对于 SNS 主题，选择您的 SNS 主题。
   b. 对于事件类型，选择要发送通知的事件。
   c. 选择 Create（创建）。

为 Auto Scaling 组配置 Amazon SNS 通知 (Amazon CLI)

使用以下 put-notification-configuration 命令。

```
aws autoscaling put-notification-configuration --auto-scaling-group-name my-
asg --topic-arn arn --notification-types "autoscaling:EC2_INSTANCE_LAUNCH"
 "autoscaling:EC2_INSTANCE_TERMINATE"
```

测试通知

要为启动事件生成通知，请通过将 Auto Scaling 组的所需容量增加 1 来更新 Auto Scaling 组。您将在实例启动后的几分钟内收到通知。

更改所需容量（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格，其中显示有关所选组的信息。
3. 在 Details（详细信息）选项卡上，选择 Group details（组详细信息）、Edit (编辑)。
4. 对于 Desired capacity (所需容量)，将当前值增加 1。如果此值超过 Maximum capacity (最大容量)，则还必须将 Maximum capacity (最大容量) 的值增加 1。
5. 选择 Update（更新）。
6. 在数分钟后，您将收到事件的通知。如果您不需要您为此测试启动的其他实例，则可以将 Desired capacity (所需容量) 减少 1。在数分钟后，您将收到事件的通知。

删除通知配置

如果您的 Amazon EC2 Auto Scaling 通知配置不再使用，您可以将其删除。

删除 Amazon EC2 Auto Scaling 通知配置（控制台）

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选择您的 Auto Scaling 组。
3. 在活动选项卡上，选中您要删除的通知旁边的复选框，然后选择操作、删除。

删除 Amazon EC2 Auto Scaling 通知配置 (Amazon CLI)

使用以下 delete-notification-configuration 命令。

```
aws autoscaling delete-notification-configuration --auto-scaling-group-name my-asg --topic-arn arn
```

有关删除 Amazon SNS 主题以及与您的 Auto Scaling 组关联的所有订阅的信息，请参阅 Amazon Simple Notification Service 开发人员指南中的删除 Amazon SNS 订阅和主题。

与 Amazon EC2 Auto Scaling 集成的 Amazon 服务

Amazon EC2 Auto Scaling 可以与其他 Amazon 服务集成。查看下面的集成选项，以详细了解各项服务如何与 Amazon EC2 Auto Scaling 配合使用。

主题

- 使用容量重新平衡来处理 Amazon EC2 竞价型实例中断 (p. 262)
- 使用 Amazon CloudFormation 创建 Auto Scaling 组 (p. 269)
- 使用 Amazon CloudShell 从命令行创建自动扩缩组 (p. 269)
- 使用 Amazon Compute Optimizer 获取 Auto Scaling 组的实例类型建议 (p. 270)
- 使用 Elastic Load Balancing 跨 Auto Scaling 组中的实例分配流量。 (p. 272)
- 使用 EventBridge 处理 Auto Scaling 事件 (p. 287)
- 使用 Amazon VPC 为 Auto Scaling 实例提供网络连接 (p. 300)

使用容量重新平衡来处理 Amazon EC2 竞价型实例中断

您可以配置 Amazon EC2 Auto Scaling，以监控和自动响应影响竞价型实例可用性的更改。容量再平衡功能可在运行中的实例被 Amazon EC2 中断之前，主动使用新竞价型实例扩展您的队列，从而帮助您维护工作负载的可用性。

工作原理

容量再平衡的目标是确保不间断地处理工作负载。如果竞价型实例处于较高的中断风险，Amazon EC2 Spot 服务将通过 EC2 实例再平衡建议通知 Amazon EC2 Auto Scaling。

为 Auto Scaling 组启用容量再平衡时，Amazon EC2 Auto Scaling 将尝试主动替换组中已收到再平衡建议的竞价型实例。这让您能够将工作负载转移到不具有较高中断风险的新竞价型实例。在 Amazon EC2 Auto Scaling 在现有实例中断之前启动新竞价型实例的同时，您的工作负载可以继续处理工作。

您还可以选择使用生命周期钩子在终止之前对实例执行自定义操作。

有关 EC2 实例再平衡建议的更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 EC2 实例再平衡建议。

有关容量再平衡功能的详细信息和演练，请参阅 Amazon 计算博客上的博客文章使用用于 EC2 Auto Scaling 的新容量再平衡功能主动管理竞价型实例生命周期。

> Note
>
> 禁用容量再平衡时，在 Amazon EC2 竞价服务中断实例且其运行状况检查失败之前，Amazon EC2 Auto Scaling 不会替换竞价型实例。在实例中断之前，Amazon EC2 始终会提供 EC2 实例再平衡建议和两分钟竞价型实例中断通知。

目录

- 注意事项 (p. 263)
  - 只有可用性相同或更好时，Amazon EC2 Auto Scaling 才会启动新实例 (p. 263)
  - 容量再平衡不会提高您的竞价型实例中断率 (p. 263)

- 启用容量再平衡 (p. 264)
  - 启用容量再平衡（控制台）(p. 264)
  - 启用容量再平衡（Amazon CLI）(p. 265)
- 添加终止生命周期钩子 (p. 267)
- 限制 (p. 268)

注意事项

容量再平衡时需考虑以下事项：

- 我们建议您将 Auto Scaling 组配置为使用多个实例类型。这样可灵活在每个可用区内的各种竞价型实例池中启动实例，如 Auto Scaling 组具有多个实例类型和购买选项 (p. 50) 中记录。
- 如果可能，您应在区域内的所有可用区中创建 Auto Scaling 组。这样，Amazon EC2 Auto Scaling 可以检查每个可用区中的可用容量。如果在一个可用区中启动失败，Amazon EC2 Auto Scaling 会继续尝试跨指定可用区启动竞价型实例，直到成功为止。
- 如果您使用 `lowest-price` 分配策略，替换竞价型实例可能会面临中断升高的风险。这是因为，即使替换竞价型实例可能在启动后不久中断，我们始终会在当时具有可用容量的价格最低池中启动实例。为避免中断风险升高，我们强烈建议不要使用 `lowest-price` 分配策略，而建议使用 `price-capacity-optimized`、`capacity-optimized` 或 `capacity-optimized-prioritized` 分配策略。这些策略旨在确保替换竞价型实例在最佳竞价型容量池中启动，因此在不久的将来不太可能发生中断。
- 使用容量再平衡时，Amazon EC2 Auto Scaling 的行为方式如下：

  新的竞价型实例启动时，Amazon EC2 Auto Scaling 会等待新实例通过其运行状况检查，然后继续终止旧实例。替换多个实例时，每个旧实例的终止将在新实例启动并通过其运行状况检查后开始。因为 Amazon EC2 Auto Scaling 尝试在终止旧实例之前启动新实例，所以达到或接近指定的最大容量可能会阻碍或完全停止再平衡活动。为避免此问题，在再平衡活动期间，Amazon EC2 Auto Scaling 可以暂时超出某组的指定最大容量。

  - 如果新实例无法启动或启动但运行状况检查失败，Amazon EC2 Auto Scaling 将继续尝试重新启动它们。当它尝试启动新实例时，您的旧实例最终将被中断并强制终止。
  - 如果某个扩缩活动正在进行并且您的 Auto Scaling 组低于其新的所需容量，则 Amazon EC2 Auto Scaling 首先横向扩展，然后再终止旧实例。
- 您可以在启用容量再平衡时为 Auto Scaling 组配置终止生命周期钩子，以确保您能够在 Amazon EC2 Auto Scaling 终止实例之前，尝试从容关闭收到再平衡通知的实例中的应用程序。如果您没有配置生命周期钩子，Amazon EC2 Auto Scaling 将在新实例通过运行状况检查后立即开始终止旧实例。

只有可用性相同或更好时，Amazon EC2 Auto Scaling 才会启动新实例

容量再平衡的目标之一是提高竞价型实例的可用性。如果现有竞价型实例收到再平衡建议，则 Amazon EC2 Auto Scaling 只有在新实例提供与现有实例相同或更好的可用性时才会启动新实例。如果新实例的中断风险比现有实例更高，则 Amazon EC2 Auto Scaling 将不会启动新实例。但是，Amazon EC2 Auto Scaling 将继续根据 Amazon EC2 竞价服务提供的信息评测竞价型容量池，并在可用性改善时启动新实例。

如果 Amazon EC2 Auto Scaling 没有主动启动新实例，则您的现有实例可能会中断。发生这种情况时，Amazon EC2 Auto Scaling 将在收到竞价型实例中断通知后立即尝试启动新实例，无论新实例中断的风险是否很高。

容量再平衡不会提高您的竞价型实例中断率

当您启用容量再平衡时，它不会提高您的竞价型实例中断率（在 Amazon EC2 需要收回容量时回收的竞价型实例的数量）。但是，如果容量再平衡检测到实例存在中断风险，Amazon EC2 Auto Scaling 将立即尝试启

动新实例。结果是，与您在中断存在风险的实例后等待 Amazon EC2 Auto Scaling 启动新实例的情况相比，可以替换更多实例。

虽然在启用了容量再平衡的情况下您可以更换更多实例，但在中断您的实例之前，您会有更多时间采取行动，从而因为处于主动而非被动地位而受益。使用 Spot Instance interruption notice（竞价型实例中断通知），您通常最多只有两分钟的时间来正常关闭您的实例。借助容量再平衡提前启动新实例，您可以使现有流程更有可能在存在风险的实例上完成，您可以启动实例关闭程序，并防止在存在风险的实例上安排新工作。您还可以开始准备新启动的实例，以接管应用程序。借助容量再平衡的主动替换，您可以因绝佳的连续性而受益。

作为演示使用容量再平衡的风险和优点的理论示例，请考虑以下场景：

- 下午 2:00 - 收到针对实例 A 的再平衡建议，Amazon EC2 Auto Scaling 立即开始尝试启动替换实例 B，让您有时间启动关闭程序。
- 下午 2:30 - 收到针对实例 B 的再平衡建议，该实例将被替换为实例 C，让您有时间启动关闭程序。
- 下午 2:32 - 如果未启用容量再平衡，并且如果在下午 2:32 收到针对实例 A 的竞价型实例中断通知，则您最多只有两分钟时间采取行动，但实例 A 将一直运行到此时。

启用容量再平衡

您可以随时启用或禁用容量再平衡。

启用容量再平衡（控制台）

您可以在创建或更新 Auto Scaling 组时启用或禁用容量再平衡。

为新的 Auto Scaling 组启用容量再平衡

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选择 Create Auto Scaling group（创建 Auto Scaling 组）。
3. 在第 1 步中，为 Auto Scaling 组输入一个名称，选择一个启动模板，然后选择 Next（下一步）以继续执行下一步骤。
4. 对于 Step 2: Choose instance launch options（第 2 步：选择实例启动选项），在 Network（网络）下选择需要的选项。
5. 对于 Instance type requirements（实例类型要求），选择设置以创建混合实例组，包括该组可以启动的实例类型、实例购买选项以及竞价型和按需型实例的分配策略。预设情况下，这些设置均未配置。要进行配置，必须选择 Override launch template（覆盖启动模板）。有关创建混合实例组的更多信息，请参阅 Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。
6. 在 Allocation strategies（分配策略）部分的页面底部，选择一个竞价型实例分配策略。要启用或禁用容量再平衡，请选中或清除 Capacity rebalance（容量再平衡）复选框。仅当您在 Instance purchase options（实例购买选项）部分指定了 Auto Scaling 组要作为竞价型实例启动的百分比时，才会看到此选项。
7. 创建 Auto Scaling 组。

为现有 Auto Scaling 组启用容量再平衡

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details（详细信息）选项卡上，依次选择 Allocation strategies（分配策略）、Edit（编辑）。

4. 在 Allocation strategies (分配策略) 中，执行以下操作：
   - 要启用容量再平衡，请选择容量再平衡复选框。
   - 要禁用容量再平衡，请清除容量再平衡复选框。
5. 选择 Update（更新）。

启用容量再平衡（Amazon CLI）

以下示例显示如何使用 Amazon CLI 以启用和禁用容量再平衡。

使用 create-auto-scaling-group 或 update-auto-scaling-group 命令以及以下参数：

- `--capacity-rebalance` / `--no-capacity-rebalance` — 指示是否启用容量再平衡的布尔值。

在您调用 create-auto-scaling-group 命令之前，您需要配置为与 Auto Scaling 组配合使用的启动模板的名称。有关更多信息，请参阅为 Auto Scaling 组创建启动模板 (p. 20)。

> **Note**
>
> 以下过程显示如何使用 JSON 或 YAML 格式的配置文件。如果您使用 Amazon CLI 版本 1，必须指定 JSON 格式的配置文件。如果您使用 Amazon CLI 版本 2，可以指定 YAML 或 JSON 格式的配置文件。

JSON

创建和配置新的 Auto Scaling 组

- 使用以下 create-auto-scaling-group 命令创建新的 Auto Scaling 组并启用容量再平衡，引用 JSON 文件作为 Auto Scaling 组的唯一参数。

```
aws autoscaling create-auto-scaling-group --cli-input-json file:///~/config.json
```

如果您还没有指定混合实例策略 (p. 50)的 CLI 配置文件，请创建一个。

将以下行添加到配置文件中的顶级 JSON 对象。

```
{
    "CapacityRebalance": true
}
```

下面是一个 `config.json` 示例文件。

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredCapacity": 12,
    "MinSize": 12,
    "MaxSize": 15,
    "CapacityRebalance": true,
    "MixedInstancesPolicy": {
        "InstancesDistribution": {
            "OnDemandBaseCapacity": 0,
            "OnDemandPercentageAboveBaseCapacity": 25,
            "SpotAllocationStrategy": "capacity-optimized"
        },
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
```

```
                "Version": "$Default"
            },
            "Overrides": [
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                },
                {
                    "InstanceType": "m5.large"
                },
                {
                    "InstanceType": "m5a.large"
                },
                {
                    "InstanceType": "c4.large"
                },
                {
                    "InstanceType": "m4.large"
                },
                {
                    "InstanceType": "c3.large"
                },
                {
                    "InstanceType": "m3.large"
                }
            ]
        }
    },
    "TargetGroupARNs": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-alb-target-group/943f017f100becff",
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

YAML

创建和配置新的 Auto Scaling 组

- 使用以下 create-auto-scaling-group 命令创建新的 Auto Scaling 组并启用容量再平衡，引用 YAML 文件作为 Auto Scaling 组的唯一参数。

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

将以下行添加到 YAML 格式的配置文件。

```
CapacityRebalance: true
```

下面是一个 `config.yaml` 示例文件。

```
---
AutoScalingGroupName: my-asg
DesiredCapacity: 12
MinSize: 12
MaxSize: 15
CapacityRebalance: true
MixedInstancesPolicy:
  InstancesDistribution:
    OnDemandBaseCapacity: 0
    OnDemandPercentageAboveBaseCapacity: 25
```

```
    SpotAllocationStrategy: capacity-optimized
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
    - InstanceType: c5.large
    - InstanceType: c5a.large
    - InstanceType: m5.large
    - InstanceType: m5a.large
    - InstanceType: c4.large
    - InstanceType: m4.large
    - InstanceType: c3.large
    - InstanceType: m3.large
TargetGroupARNs:
- arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-alb-target-
group/943f017f100becff
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

为现有 Auto Scaling 组启用容量再平衡

- 使用以下 update-auto-scaling-group 命令启用容量再平衡。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --capacity-rebalance
```

验证是否为 Auto Scaling 组启用容量再平衡

- 使用以下 describe-auto-scaling-groups 命令验证是否启用了容量再平衡并查看详细信息。

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

以下为响应示例。

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn",
            ...
            "CapacityRebalance": true
        }
    ]
}
```

禁用容量再平衡

使用包含 `--no-capacity-rebalance` 选项的 update-auto-scaling-group 命令以禁用容量再平衡。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --no-capacity-rebalance
```

添加终止生命周期钩子

您还可以在启用容量重新平衡时配置终止生命周期钩子。借助生命周期钩子，您可以在实例终止之前在实例上执行自定义操作。

以下是可能需要使用终止生命周期钩子的一些原因：

- 将系统或应用程序日志上载到 Amazon Simple Storage Service (Amazon S3)
- 用于正常关闭 Amazon SQS 工件
- 完成从域名系统 (DNS) 取消注册

如果您还没有终止生命周期钩子，请使用以下步骤创建。

添加终止生命周期钩子

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中您的 Auto Scaling 组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Instance management (实例管理) 选项卡的 Lifecycle hooks (生命周期挂钩) 中，选择 Create lifecycle hook (创建生命周期挂钩)。
4. 要定义生命周期挂钩，请执行以下操作：

   a. 对于 Lifecycle hook name (生命周期挂钩名称)，请指定生命周期挂钩的名称。

   b. 对于生命周期转换，请选择实例终止。

   c. 对于检测信号超时中，指定完成生命周期操作所需的时间量（以秒为单位），或直到超时期结束。我们建议您使用 30 秒到 120 秒的值，具体取决于关闭应用程序需要的时间。

   d. 对于 Default result（默认结果），请指定超时结束或发生意外故障时 Auto Scaling 组执行的操作。ABANDON（放弃）和 CONTINUE（继续）都允许终止实例。

      - 如果您选择 CONTINUE（继续），Auto Scaling 组可以在终止之前继续执行任何剩余操作（如其他生命周期钩子）。
      - 如果选择 ABANDON（放弃），Auto Scaling 组将立即终止实例。

   e. （可选）对于 Notification metadata（通知元数据），请指定您希望 Amazon EC2 Auto Scaling 每次在向您在第 6 步中配置的 Amazon Lambda 函数或其他通知目标发送消息时包含的其他信息。
5. 选择 Create（创建）。
6. （可选）要在实例终止之前使用 Lambda 之类的服务执行自定义操作，请参阅教程：配置调用 Lambda 函数的生命周期钩子 (p. 207)。本教程将帮助您了解如何为生命周期钩子设置 Lambda 函数。否则，要使 EC2 实例自动运行操作，您必须将其配置为运行关机脚本。我们建议您编写一个可在一到两分钟内完成的关闭顺序脚本，以确保在实例终止之前有足够的时间完成任务。

要了解更多信息以帮助您了解有关生命周期钩子用法的不同方面，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

限制

- Amazon EC2 Auto Scaling 只有在接收再平衡通知的实例未受横向缩减保护的情况下才能终止该实例。
- 您的应用程序应该能够处理竞价型实例提前中断的可能性。例如，当实例开始终止时，Amazon EC2 Auto Scaling 将等待实例终止。如果 Auto Scaling 组位于 Elastic Load Balancing 负载均衡器后面，则 Amazon EC2 Auto Scaling 会等待实例从负载均衡器中取消注册，然后再调用终止生命周期钩子（如果已配置）。如果注销实例和完成生命周期操作花费的时间过长，则实例可能会在 Amazon EC2 Auto Scaling 等待实例终止期间中断。

  Amazon EC2 也并不总能在两分钟的竞价型实例中断通知之前发送再平衡建议信号。在某些情况下，再平衡建议信号可能会随两分钟的中断通知一起到达。

如果实例收到最后两分钟的中断通知，Amazon EC2 Auto Scaling 会调用终止生命周期钩子并尝试立即启动替换。

- 所有提供 Amazon EC2 Auto Scaling 的商业 Amazon Web Services 区域 均支持容量再平衡，但不包括中东（阿联酋）区域。

使用 Amazon CloudFormation 创建 Auto Scaling 组

Amazon EC2 Auto Scaling 与 Amazon CloudFormation 服务集成，后者可帮助您对 Amazon 资源进行建模和设置，从而节省创建和管理资源与基础设施所需的时间。您可以创建一个描述所需的全部 Amazon 资源（例如 Auto Scaling 组）的模板，然后 Amazon CloudFormation 将为您预置和配置这些资源。

在您使用 Amazon CloudFormation 时，可重复使用您的模板来不断地重复设置您的 Amazon EC2 Auto Scaling 资源。描述您的资源一次，然后在多个 Amazon Web Services 账户 和区域中反复预置相同的资源。

Amazon EC2 Auto Scaling 和 Amazon CloudFormation 模板

要为 Amazon EC2 Auto Scaling 以及相关服务预置和配置资源，您必须了解 Amazon CloudFormation 模板。模板是 JSON 或 YAML 格式的文本文件。这些模板描述要在 Amazon CloudFormation 堆栈中调配的资源。如果您不熟悉 JSON 或 YAML，可以在 Amazon CloudFormation Designer 的帮助下开始使用 Amazon CloudFormation 模板。有关更多信息，请参阅 Amazon CloudFormation 用户指南中的什么是 Amazon CloudFormation Designer？。

要开始为 Amazon EC2 Auto Scaling 创建自己的堆栈模板，请完成以下任务：

- 使用 AWS::EC2::LaunchTemplate 创建启动模板。
- 使用 AWS::AutoScaling::AutoScalingGroup 创建自动扩缩组。

有关演示如何在应用程序负载均衡器后部署自动扩缩组的演练，请参阅《Amazon CloudFormation 用户指南》中的演练：创建经扩展和负载均衡的应用程序。

有关创建自动扩缩组和相关资源的模板的更多有用示例，请参阅《Amazon CloudFormation 用户指南》的 Auto scaling 模板代码段部分。有关更多信息和示例代码段，请参阅《Amazon CloudFormation 用户指南》中的 Amazon EC2 Auto Scaling 资源类型参考。

了解有关 Amazon CloudFormation 的更多信息

要了解有关 Amazon CloudFormation 的更多信息，请参阅以下资源：

- Amazon CloudFormation
- Amazon CloudFormation 用户指南
- Amazon CloudFormation API 参考
- Amazon CloudFormation 命令行界面用户指南

使用 Amazon CloudShell 从命令行创建自动扩缩组

在支持的 Amazon Web Services 区域中，您可以使用 Amazon CloudShell 运行 Amazon CLI 命令，以实现可从 Amazon Web Services Management Console 直接启动并且已经预先通过身份验证的浏览器式 shell。您可以运行 Amazon CLI 命令对服务使用您的首选 shell（Bash、PowerShell 或 Z shell）。

您可以用以下两种方法之一从 Amazon Web Services Management Console 启动 Amazon CloudShell：

- 选择控制台导航栏中的 Amazon CloudShell 图标。它位于搜索框的右侧。
- 使用控制台导航栏上的搜索框搜索 CloudShell，然后选择 CloudShell 选项。

首次在新的浏览器窗口中启动 Amazon CloudShell 时，欢迎面板将显示并列出主要功能。关闭此面板后，系统会在 shell 配置和转发控制台凭证的同时提供状态更新。当系统显示命令提示符时，表示 shell 已经准备就绪，可以进行交互。

有关此服务的更多信息，请参阅 Amazon CloudShell 用户指南。

使用 Amazon Compute Optimizer 获取 Auto Scaling 组的实例类型建议

Amazon 提供 Amazon EC2 实例建议，通过使用 Amazon Compute Optimizer 支持的功能，帮助您提高性能、节省资金或实现这两点。您可以根据这些建议来决定是否移动到新的实例类型。

为了生成建议，Compute Optimizer 会分析现有实例规范和最近指标历史记录。然后，利用编译数据提供已得到最佳优化的 Amazon EC2 实例类型的建议，以处理现有性能工作负载。建议随每小时实例定价一起返回。

> Note
>
> 要从 Compute Optimizer 中获取建议，您必须首先选择加入 Compute Optimizer。有关更多信息，请参阅 Amazon Compute Optimizer 用户指南中的 Amazon Compute Optimizer 入门。

目录

- 限制 (p. 270)
- 结果 (p. 270)
- 查看建议 (p. 271)
- 评估建议时的注意事项 (p. 271)

限制

Compute Optimizer 为 Auto Scaling 组中的实例生成建议，这些实例配置为启动和运行 M、C、R、T 和 X 实例类型。但是，它不会为 Amazon Graviton2 处理器（例如 C6g）支持的 -g 实例类型以及具有更高网络带宽性能（例如 M5n）的 -n 实例类型生成建议。

Auto Scaling 组还必须配置为运行单个实例类型（即无混合实例类型），不得附加扩展策略，并且对于所需容量、最小容量和最大容量（即具有固定数量实例的 Auto Scaling 组）具有相同的值。Compute Optimizer 为 Auto Scaling 组中满足所有这些配置要求的实例生成建议。

结果

Compute Optimizer 对 Auto Scaling 组的结果进行分类，如下所示：

- 未优化 – 当 Compute Optimizer 确定可为您的工作负载提供更好性能的建议时，Auto Scaling 组被视为未优化。
- 优化 – 当 Compute Optimizer 确定已根据所选实例类型正确预配置 Auto Scaling 组以运行工作负载时，该组将被视为已优化。对于优化的资源，Compute Optimizer 有时可能会建议新一代实例类型。
- 无 – 没有关于该 Auto Scaling 组的建议。如果您选择使用 Computer Optimizer 的时间少于 12 小时、Auto Scaling 组的运行时间少于 30 小时，或者 Computer Optimizer 不支持 Auto Scaling 组或实例类型，则可能会发生这种情况。想要了解更多信息，请参阅 限制 (p. 270) 部分。

查看建议

选择使用 Compute Optimizer 后，您可以查看它为 Auto Scaling 组生成的结果和建议。如果您是最近选择使用的，可能在长达 12 小时内不会提供建议。

查看为 Auto Scaling 组生成的建议

1. 打开 Compute Optimizer 控制台，网址为 https://console.aws.amazon.com/compute-optimizer/。

   此时将打开“控制面板”页面。
2. 选择 View recommendations for all Auto Scaling groups（查看所有 Auto Scaling 组的建议）。
3. 选择您的 Auto Scaling 组。
4. 选择 View detail（查看详细信息）。

   视图根据默认表设置在预配置视图中更改为最多显示三种不同的实例建议。它还提供 Auto Scaling 组的最新 CloudWatch 指标数据（平均 CPU 利用率、平均网络输入和平均网络输出）。

确定是否要使用其中某个建议。决定是否要进行优化以便提高性能和/或节省资金。

要更改 Auto Scaling 组中的实例类型，请更新启动模板或更新 Auto Scaling 组以使用新的启动配置。现有实例将继续使用以前的配置。要更新现有实例，请终止这些实例，以便 Auto Scaling 组替换这些实例，或者根据您的终止策略 (p. 222)启用自动扩展，以逐步使用较新实例替换较旧实例。

> Note
>
> 借助最长实例生命周期和实例刷新功能，您还可以替换 Auto Scaling 组中的现有实例，以启动使用新启动模板或启动配置的新实例。有关更多信息，请参阅 基于最大实例生命周期替换 Auto Scaling 实例 (p. 115) 和 基于实例刷新替换 Auto Scaling 实例 (p. 95)。

评估建议时的注意事项

移至新的实例类型时，请考虑以下事项：

- 这些建议不会预测您的使用情况。建议基于您在最近 14 天时间段内的历史使用情况。请务必选择一种预计能够满足您的未来使用需求的实例类型。
- 关注图表指标以确定实际使用量是否低于实例容量。您还可以在 CloudWatch 中查看指标数据（平均值、峰值、百分比），以进一步评估 EC2 实例建议。例如，观察当天 CPU 百分比指标如何变化，以及是否有需要满足的峰值。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的查看可用指标。
- Compute Optimizer 可能会为可突增性能实例（即 T3、T3a 和 T2 实例）提供建议。如果您定期突增到基准以上，请确保您可以根据新实例类型的 vCPU 继续执行此操作。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的可突增性能实例的 CPU 积分和基准性能。
- 如果您购买了 Reserved Instance，您的按需型实例可能会作为预留实例进行计费。在更改当前实例类型之前，请首先评估对Reserved Instance使用率和覆盖率的影响。
- 尽可能考虑转换为较新一代实例。
- 在迁移到其他实例系列时，请确保当前实例类型和新实例类型在虚拟化、架构或网络类型等方面兼容。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的调整实例大小的兼容性。
- 最后，请考虑为每个建议提供的性能风险评级。性能风险指示您为了验证建议的实例类型是否满足工作负载的性能要求而可能需要执行的工作量。我们还建议在进行任何更改前后进行严格的负载和性能测试。

其他资源

除了本页面上的主题以外，还可以参阅以下资源：

- Amazon EC2 实例类型

- Amazon Compute Optimizer 用户指南

使用 Elastic Load Balancing 跨 Auto Scaling 组中的实例分配流量。

Elastic Load Balancing 可让您自动在运行的所有 EC2 实例之间分配传入的应用程序流量。Elastic Load Balancing 通过最佳路由流量管理传入请求，以便所有实例都不会负载过多。

要将 Elastic Load Balancing 与您的 Auto Scaling 组配合使用，请将负载均衡器附加到您的 Auto Scaling 组 (p. 274)。这将使用负载均衡器注册该组，负载均衡器将作为到您的 Auto Scaling 组的所有传入 Web 流量的单一接触点。

当您将 Elastic Load Balancing 与您的 Auto Scaling 组配合使用时，无需使用负载均衡器注册单独的 EC2 实例。您的 Auto Scaling 组所启动的实例会自动注册到负载均衡器。同样，您的 Auto Scaling 组所终止的实例会自动从负载均衡器取消注册。

将负载均衡器附加到您的 Auto Scaling 组后，可以将您的 Auto Scaling 组配置为使用 Elastic Load Balancing 指标（如每个目标的 Application Load Balancer 请求计数）来随着需求波动而扩展该组中的实例数量。

您可以选择将 Elastic Load Balancing 运行状况检查添加到您的 Auto Scaling 组，以便 Amazon EC2 Auto Scaling 可以根据这些额外的运行状况检查识别和替换运行状况不佳的实例。否则，您可以创建 CloudWatch 警报，该警报在目标组的运行状况良好的主机计数低于允许的情况下会通知您。

目录

- Elastic Load Balancing 类型 (p. 272)
- Elastic Load Balancing 入门先决条件 (p. 273)
- 将负载均衡器附加到 Auto Scaling 组 (p. 274)
- 了解负载均衡器的附加状态 (p. 275)
- 从 Amazon EC2 Auto Scaling 控制台配置 Application Load Balancer 或 Network Load Balancer。 (p. 276)
- 向 Auto Scaling 组添加 Elastic Load Balancing 运行状况检查 (p. 276)
- 添加和移除可用区 (p. 277)
- 将 Elastic Load Balancing 与 Amazon Command Line Interface (Amazon CLI) 配合使用的示例 (p. 279)
- 教程：设置具有扩展和负载均衡功能的应用程序 (p. 282)

Elastic Load Balancing 类型

Elastic Load Balancing 提供四种类型的负载均衡器，可与您的 Auto Scaling 组配合使用：Application Load Balancer、Network Load Balancer、网关负载均衡器和经典负载均衡器。

负载均衡器类型的配置方式具有一个关键区别。通过 Application Load Balancer、Network Load Balancer 和网关负载均衡器，可以使用目标组将实例注册为目标并将流量路由到目标组。通过经典负载均衡器，可以使用负载均衡器直接注册实例。

Application Load Balancer

路由和负载均衡在应用程序层 (HTTP/HTTPS) 进行，并支持基于路径的路由。Application Load Balancer 可以将请求路由到一个或多个注册目标上的端口，例如，Virtual Private Cloud (VPC) 中的 EC2 实例。

Network Load Balancer

路由和负载均衡在传输层（TCP/UDP 第 4 层）进行，依据是从第 4 层中提取的地址信息。Network Load Balancer 可以处理突发流量，保留客户端的源 IP，并在负载均衡器的使用寿命内使用固定 IP。

网关负载均衡器

将流量分配到设备实例队列。为第三方虚拟设备（如防火墙、入侵检测和防御系统以及其他设备）提供可扩展性、可用性和简单性。网关负载均衡器与支持 GENEVE 协议的虚拟设备配合使用。需要额外的技术集成，因此请务必在选择网关负载均衡器之前参考用户指南。

经典负载均衡器

路由和负载均衡在传输层 (TCP/SSL) 或在应用程序层 (HTTP/HTTPS) 进行。

要了解有关 Elastic Load Balancing 的更多信息，请参阅以下主题：

- 什么是 Elastic Load Balancing？
- 什么是 Application Load Balancer？
- 什么是 Network Load Balancer？
- 什么是网关负载均衡器？
- 什么是经典负载均衡器？

Elastic Load Balancing 入门先决条件

您必须完成先决条件，然后将负载均衡器附加到您的自动扩缩组。这包括创建负载均衡器和用于将流量路由到您的自动扩缩组的目标组。

按照 Elastic Load Balancing 文档中的过程创建负载均衡器和目标组。跳过注册 Amazon EC2 实例的步骤。Amazon EC2 Auto Scaling 将自动负责注册（和取消注册）实例。有关更多信息，请参阅 Elastic Load Balancing 用户指南中的 Elastic Load Balancing 入门。

或者，如果要创建 Application Load Balancer 或 Network Load Balancer，则无需立即创建负载均衡器和目标组。您可以从 Amazon EC2 Auto Scaling 控制台创建和附加新的 Application Load Balancer 或 Network Load Balancer。有关更多信息，请参阅从 Amazon EC2 Auto Scaling 控制台配置 Application Load Balancer 或 Network Load Balancer。 (p. 276)。

要将某个负载均衡器附加到您的自动扩缩组，请首先确保您满足以下要求：

- 负载均衡器及其目标组必须与您的自动扩缩组位于同一 Amazon Web Services 账户、VPC 和区域。
- 目标组必须指定的 `instance` 目标类型。使用 Auto Scaling 组时，无法指定 `ip` 的目标类型。
- 必须为您的启动模板或启动配置设置安全组，以允许来自该负载均衡器的流量到达您的 Auto Scaling 组。推荐规则取决于负载均衡器的类型和负载均衡器使用的后端类型。例如，要将流量路由到 Web 服务器，请允许从负载均衡器在端口 80 上进行入站 HTTP 访问。
- 在网关负载均衡器后部署虚拟设备时，Amazon Machine Image（AMI）必须指定某个支持 GENEVE 协议的 AMI 的 ID，以允许 Auto Scaling 组与网关负载均衡器交换流量。此外，您在启动模板或启动配置中指定的安全组必须允许 UDP 流量通过端口 6081。

在允许您的自动扩缩组使用 Elastic Load Balancing 运行状况检查的结果来确定实例的运行状况之前，请确保您已满足此附加要求：

- 您在启动模板或启动配置中指定的安全组必须允许从该 Elastic Load Balancing 负载均衡器通过正确的端口访问以执行运行状况检查。有关更多信息，请参阅向 Auto Scaling 组添加 Elastic Load Balancing 运行状况检查 (p. 276)。

Tip

如果您拥有需要一段时间才能完成的引导启动脚本，则可以选择向您的 Auto Scaling 组添加启动生命周期钩子，以便在引导启动脚本成功完成并且实例上的应用程序准备好接受流量之前，推迟将实例注册到负载均衡器后。首次在 Amazon EC2 Auto Scaling 控制台中创建 Auto Scaling 组时，您将无法添加生命周期钩子。您可以在创建组后再添加生命周期钩子。有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

将负载均衡器附加到 Auto Scaling 组

本主题介绍如何将 Elastic Load Balancing 负载均衡器连接到您的 Auto Scaling 组。Amazon EC2 Auto Scaling 与 Elastic Load Balancing 集成，可帮助您在 Auto Scaling 组前面插入 Application Load Balancer、Network Load Balancer、经典负载均衡器或网关负载均衡器。要了解有关不同类型的负载均衡器的更多信息，请参阅 Elastic Load Balancing 类型 (p. 272)。

当您附加 Application Load Balancer、Network Load Balancer 或网关负载均衡器时，您将附加目标组。Amazon EC2 Auto Scaling 在实例启动时将其添加到附加的目标组。您可以附加一个或多个目标组，并配置每个目标组的运行状况检查。

有关将目标组附加到 Auto Scaling 组的介绍指南，请参阅教程：设置具有扩展和负载均衡功能的应用程序 (p. 282)。

在开始本过程之前，请完成先决条件 (p. 273)。

目录

- 附加现有负载均衡器 (p. 274)
- 分离负载均衡器 (p. 275)

附加现有负载均衡器

创建或更新组时，可以将现有负载均衡器附加到 Auto Scaling 组。如果要在创建组的同时创建和附加新的 Application Load Balancer 或 Network Load Balancer，请参阅 从 Amazon EC2 Auto Scaling 控制台配置 Application Load Balancer 或 Network Load Balancer。 (p. 276)。

在创建新的 Auto Scaling 组时附加现有负载均衡器

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选择 Create Auto Scaling group（创建 Auto Scaling 组）。
3. 在步骤 1 和 2中，选择所需的选项，然后继续执行步骤 3：配置高级选项。
4. 对于负载均衡，选择附上现有负载均衡器。
5. 在附加到现有负载均衡器，请执行以下操作之一：

   a. 对于 Application Load Balancer、Network Load Balancer 和网关负载均衡器：

      选择从负载均衡器目标组中选择，然后在现有负载均衡器目标组字段中选择目标组。

   b. 对于经典负载均衡器：

      选择从经典负载均衡器中选择，然后在 经典负载均衡器 字段中选择您的负载均衡器。
6. 继续创建 Auto Scaling 组。创建 Auto Scaling 组后，您的实例将自动注册到负载均衡器。

将现有负载均衡器附加到现有 Auto Scaling 组

使用以下过程将负载均衡器附加到现有的 Auto Scaling 组。

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中现有组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details (详细信息) 选项卡上，选择 Load balancing (负载均衡)、Edit (编辑)。
4. 在 Load balancing (负载均衡) 下，执行下列操作之一：

   a. 对于应用程序、网络或网关负载均衡器目标组，选中其复选框，然后选择一个目标组。

   b. 对于 经典负载均衡器，选中其复选框，然后选择您的负载均衡器。
5. 选择 Update（更新）。

   Note

   在附加负载均衡器时，您可以使用 Amazon CLI 来监控负载均衡器的状态。当 Amazon EC2 Auto Scaling 成功注册实例并且至少有一个已注册的实例通过了运行状况检查时，您会收到 `InService` 状态。有关更多信息，请参阅了解负载均衡器的附加状态 (p. 275)。

分离负载均衡器

如果不再需要负载均衡器，请使用以下步骤将其与 Auto Scaling 组分离。

将负载均衡器与组分离

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中现有组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details (详细信息) 选项卡上，选择 Load balancing (负载均衡)、Edit (编辑)。
4. 在 Load balancing (负载均衡) 下，执行下列操作之一：

   a. 对于应用程序、网络或网关负载均衡器目标组，请选择目标组旁边的删除 (X) 图标。

   b. 对于 经典负载均衡器，请选择负载均衡器旁边的删除 (X) 图标。
5. 选择 Update（更新）。

了解负载均衡器的附加状态

您可以使用 Amazon CLI 来获取负载均衡器的附加状态。有关更多信息，请参阅《Amazon CLI 命令参考》中的 describe-load-balancer-target-groups 和 describe-load-balancers 文档。

当您使用 describe-load-balancer-target-groups 命令描述目标组或使用 describe-load-balancers 命令描述负载均衡器时，Amazon EC2 Auto Scaling 会在 `State` 参数中返回负载均衡器的附加状态。

当您附加负载均衡器时，它进入 `Adding` 状态，同时注册组中的实例。注册了组中的所有实例后，它进入 `Added` 状态。在至少一个注册实例通过运行状况检查后，它进入 `InService` 状态。当负载均衡器进入 `InService` 状态时，Amazon EC2 Auto Scaling 可以终止和替换报告为运行状况不佳的所有实例。如果注册的实例均未通过运行状况检查（例如，由于未正确配置运行状况检查），负载均衡器不会进入 `InService` 状态。Amazon EC2 Auto Scaling 不会终止和替换这些实例。

当分离负载均衡器时，它进入 `Removing` 状态，同时取消注册组中的实例。实例在取消注册后仍保持运行。预设情况下，Application Load Balancer、Network Load Balancer 和 Gateway Load Balancer 启用 Connection Draining（注销延迟）功能。如果启用了 Connection Draining，则 Elastic Load Balancing 将等待动态请求完成或最大超时到期（以先到者为准），然后再取消注册实例。

从 Amazon EC2 Auto Scaling 控制台配置 Application Load Balancer 或 Network Load Balancer。

在创建您的 Auto Scaling 组时，使用以下过程创建和附加 Application Load Balancer 或 Network Load Balancer。

在创建新的 Auto Scaling 组时创建和附加新的负载均衡器

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选择 Create Auto Scaling group（创建 Auto Scaling 组）。
3. 在步骤 1 和 2 中，选择所需的选项，然后继续执行步骤 3：配置高级选项。
4. 对于负载均衡，选择附加到新的负载均衡器。
   a. 在 Attach to a new load balancer（附加到新的负载均衡器）下，对于 Load balancer type（负载均衡器类型），选择是创建 Application Load Balancer 还是 Network Load Balancer。
   b. 对于负载均衡器名称，输入负载均衡器的名称，或者保留默认名称。
   c. 对于负载均衡器模式，选择是创建面向互联网的公共负载均衡器，还是保留内部负载均衡器的默认负载均衡器。
   d. 对于可用区和子网，为在其中启动 EC2 实例的每个可用区选择公有子网。（这些都是从步骤 2 预填充的。）
   e. 对于侦听器和路由，请更新侦听器的端口号（如有必要），然后在默认路由中，选择创建目标组。或者，您可以从下拉列表中选择现有目标组。
   f. 如果您选择在最后一步创建目标组，对于新目标组名称，请输入目标组的名称，或者保留默认名称。
   g. 要为负载均衡器添加标签，请选择 Add tag（添加标签），然后提供每个标签的标签键和值。
5. 继续创建 Auto Scaling 组。创建 Auto Scaling 组后，您的实例将自动注册到负载均衡器。

   Note

   创建 Auto Scaling 组后，您可以使用 Elastic Load Balancing 控制台创建其他侦听器。如果您需要使用安全协议（如 HTTPS，或 UDP 侦听器）创建侦听器，这将非常有用。只要使用不同的端口，您就可以向现有负载均衡器添加更多侦听器。

向 Auto Scaling 组添加 Elastic Load Balancing 运行状况检查

Auto Scaling 组的默认运行状况检查只有 EC2 状态检查。如果某个实例未通过这些状态检查，则该实例将被标记为运行状况不佳并在 Amazon EC2 Auto Scaling 启动新的替换实例时终止。

您可以将一个或多个负载均衡器目标组、一个或多个经典负载均衡器或二者附加到 Auto Scaling 组。但预设情况下，Auto Scaling 组不会认为实例运行状况不佳，并会在实例未通过 Elastic Load Balancing 运行状况检查时将其替换。

为了确保您的 Auto Scaling 组可以根据其他负载均衡器测试确定实例运行状况，请将 Auto Scaling 组配置为使用 Elastic Load Balancing (ELB) 运行状况检查。负载均衡器会定期发送 ping、尝试进行连接或者发送请求来测试 EC2 实例并确定实例运行状况是否不佳。如果您将 Auto Scaling 组配置为使用 Elastic Load Balancing 运行状况检查，而它未能通过 EC2 状态检查或 Elastic Load Balancing 运行状况检查，则它会认为该实例运行状况不佳。如果您将多个负载均衡器目标组或经典负载均衡器附加到该组，则只有在所有负载均衡器目标组或经典负载均衡器均报告某实例运行状况良好的情况下，它才会认为该实例运行状况良好。如果其中任何一个负载均衡器目标组或经典负载均衡器将实例报告为运行状况不佳，则 Auto Scaling 组将替换该实例，即使其他负载均衡器目标组或经典负载均衡器将实例报告为运行状况良好也是如此。

添加 Elastic Load Balancing 运行状况检查

使用以下步骤将 Elastic Load Balancing (ELB) 运行状况检查添加到 Auto Scaling 组。

添加 Elastic Load Balancing 运行状况检查

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中现有组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details (详细信息) 选项卡上，选择 Health checks (运行状况检查)、Edit (编辑)。
4. 对于 Health check type (运行状况检查类型)，选择 Enable ELB health checks (启用 ELB 运行状况检查)。
5. 对于 Health check grace period（运行状况检查宽限期），输入 Amazon EC2 Auto Scaling 在实例进入 `InService` 状态后检查实例的运行状况之前需要等待的时间（以秒为单位）。有关更多信息，请参阅设置自动扩缩组的运行状况检查宽限期 (p. 246)。
6. 选择 Update（更新）。
7. 在 Instance management (实例管理) 选项卡上的 Instances (实例) 下，您可以查看实例的运行状况。Health Status (运行状况) 列显示新添加的运行状况检查的结果。

另请参阅

- 有关更多信息，请参阅Auto Scaling 实例的运行状况检查 (p. 241)。
- 要为您的 Application Load Balancer 配置运行状况检查，请参阅适用于 Application Load Balancer 的用户指南中的目标组的运行状况检查。
- 要为您的 Network Load Balancer 配置运行状况检查，请参阅《Network Load Balancer 用户指南》中的目标组的运行状况检查。
- 要为您的网关负载均衡器配置运行状况检查，请参阅适用于网关负载均衡器的用户指南中的目标组的运行状况检查。
- 要为您的经典负载均衡器配置运行状况检查，请参阅适用于经典负载均衡器的用户指南中的为您的经典负载均衡器配置运行状况检查。

添加和移除可用区

为利用地理冗余的安全性和可靠性，请使自动扩缩组跨您工作所在区域中的多个可用区，然后附加负载均衡器以跨这些可用区分配传入流量。

当一个可用区运行状况不佳或无法使用时，Amazon EC2 Auto Scaling 将在不受影响的可用区中启动新实例。当运行状况不佳的可用区恢复运行状况时，Amazon EC2 Auto Scaling 会自动在 Auto Scaling 组的所有可用区中重新均匀分配应用程序实例。Amazon EC2 Auto Scaling 通过尝试在实例最少的可用区中启动新实例来执行此操作。但是，如果尝试失败，Amazon EC2 Auto Scaling 将尝试在其他可用区中启动，直到成功为止。

Elastic Load Balancing 会为您为负载均衡器启用的每个可用区创建一个负载均衡器节点。如果您为负载均衡器启用了跨区域负载均衡，则每个负载均衡器节点会在所有启用的可用区中的已注册目标之间平均分配流量。如果禁用了跨区域负载均衡，则每个负载均衡器节点会仅在其可用区中的已注册实例之间平均分配请求。

创建 Auto Scaling 组时，必须指定至少一个可用区。之后，您可以通过将可用区添加到您的 Auto Scaling 组中，然后为您的负载均衡器启用该可用区（如果负载均衡器支持），来扩展应用程序的可用性。

目录

- 添加可用区 (p. 278)
- 删除可用区 (p. 278)
- 限制 (p. 279)

添加可用区

使用以下过程将 Auto Scaling 组和负载均衡器扩展到其他可用区中的子网。

添加可用区

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中现有组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details (详细信息) 选项卡上，选择 Network (网络)、Edit (编辑)。
4. 在子网中，选择与要添加到 Auto Scaling 组的可用区相对应的子网。
5. 选择 Update（更新）。
6. 要更新负载均衡器的可用区，使其与您的 Auto Scaling 组共享相同的可用区，请完成以下步骤：

   a. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
   b. 选择负载均衡器。
   c. 请执行下列操作之一：
      - 对于 Application Load Balancer 和 Network Load Balancer：

        1. 在描述选项卡上，为可用区选择编辑子网。
        2. 在编辑子网页面上，为可用区选中要添加的可用区的复选框。如果该区域只有一个子网，则将选择此子网。如果该区域有多个子网，请选择其中一个子网。
      - 对于 VPC 中的经典负载均衡器：

        1. 在 Instances 选项卡中，选择 Edit Availability Zones。
        2. 在添加和删除子网页面上，对于可用子网，使用其添加 (+) 图标选择该子网。该子网将移到 Selected subnets 下。
   d. 选择 Save（保存）。

删除可用区

要从您的 Auto Scaling 组和负载均衡器中删除可用区，请使用以下步骤。

删除可用区

1. 访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，然后从导航窗格中选择 Auto Scaling Groups（Auto Scaling 组）。
2. 选中现有组旁边的复选框。

   这时将在 Auto Scaling groups（Auto Scaling 组）页面底部打开一个拆分窗格。
3. 在 Details (详细信息) 选项卡上，选择 Network (网络)、Edit (编辑)。
4. 在子网中，选择与您要从 Auto Scaling 组删除的可用区相对应的子网的删除 (X) 图标。如果该区域有多个子网，请为每个子网选择删除 (X) 图标。
5. 选择 Update（更新）。
6. 要更新负载均衡器的可用区，使其与您的 Auto Scaling 组共享相同的可用区，请完成以下步骤：

a. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
b. 选择负载均衡器。
c. 请执行下列操作之一：
   - 对于 Application Load Balancer 和 Network Load Balancer：
     1. 在描述选项卡上，为可用区选择编辑子网。
     2. 在编辑子网页面上，为可用区清除要删除该可用区子网的复选框。
   - 对于 VPC 中的经典负载均衡器：
     1. 在 Instances 选项卡中，选择 Edit Availability Zones。
     2. 在添加和删除子网页面上，对于可用子网，使用其删除 (-) 图标删除该子网。子网将移至可用子网下。
d. 选择 Save（保存）。

限制

要更新为负载均衡器启用的可用区，您需要了解以下限制：

- 如果为您的负载均衡器启用可用区，请指定该可用区中的一个子网。请注意，您最多可为负载均衡器启用每个可用区最多一个子网。
- 对于面向互联网的负载均衡器，您为负载均衡器指定的子网必须至少具有八个可用 IP 地址。
- 对于 Application Load Balancer，您必须启用至少两个可用区。
- 对于 Network Load Balancer，您无法禁用已启用的可用区，但可以启用其他可用区。
- 对于网关负载均衡器，您无法更改创建负载均衡器时添加的可用区或子网。

将 Elastic Load Balancing 与 Amazon Command Line Interface (Amazon CLI) 配合使用的示例

使用 Amazon CLI 附加和分离负载均衡器、添加 Elastic Load Balancing 运行状况检查并更新可用区。

目录

- 附加负载均衡器目标组 (p. 279)
- 描述负载均衡器目标组 (p. 280)
- 分离负载均衡器目标组 (p. 280)
- 附加经典负载均衡器 (p. 280)
- 描述经典负载均衡器 (p. 280)
- 分离经典负载均衡器 (p. 280)
- 添加 Elastic Load Balancing 运行状况检查 (p. 281)
- 更新可用区 (p. 281)

附加负载均衡器目标组

以下 create-auto-scaling-group 命令将创建一个带附加目标组的 Auto Scaling 组。为 Application Load Balancer、Network Load Balancer 或网关负载均衡器指定目标组的 Amazon Resource Name (ARN)。

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-template "LaunchTemplateName=my-launch-template,Version=1" \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
```

```
  --target-group-arns "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-
targets/1234567890123456" \
  --max-size 5 --min-size 1 --desired-capacity 2
```

以下 attach-load-balancer-target-groups 命令将目标组附加到现有 Auto Scaling 组。

```
aws autoscaling attach-load-balancer-target-groups --auto-scaling-group-name my-asg \
  --target-group-arns "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-
targets/1234567890123456"
```

描述负载均衡器目标组

要查看与 Auto Scaling 组关联的目标组，请使用 describe-load-balancer-target-groups 命令。以下示例列出了 *my-asg* 的目标组。

```
aws autoscaling describe-load-balancer-target-groups --auto-scaling-group-name my-asg
```

有关输出中的 State 字段说明，请参阅上一个主题中的 了解负载均衡器的附加状态 (p. 275) 部分。

分离负载均衡器目标组

不再需要时，以下 detach-load-balancer-target-groups 命令将目标组与您的 Auto Scaling 组分离。

```
aws autoscaling detach-load-balancer-target-groups --auto-scaling-group-name my-asg \
  --target-group-arns "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-
targets/1234567890123456"
```

附加经典负载均衡器

以下 create-auto-scaling-group 命令将创建一个带附加经典负载均衡器的 Auto Scaling 组。

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \
  --launch-configuration-name my-launch-config \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \
  --load-balancer-names "my-load-balancer" \
  --max-size 5 --min-size 1 --desired-capacity 2
```

以下 attach-load-balancers 命令将指定的经典负载均衡器附加到现有 Auto Scaling 组。

```
aws autoscaling attach-load-balancers --auto-scaling-group-name my-asg \
  --load-balancer-names my-lb
```

描述经典负载均衡器

要查看与 Auto Scaling 组关联的经典负载均衡器，请使用 describe-load-balancers 命令。以下示例列出了 *my-asg* 的经典负载均衡器。

```
aws autoscaling describe-load-balancers --auto-scaling-group-name my-asg
```

有关输出中的 State 字段说明，请参阅 了解负载均衡器的附加状态 (p. 275)。

分离经典负载均衡器

不再需要时，以下 detach-load-balancers 命令将经典负载均衡器与 Auto Scaling 组分离。

```
aws autoscaling detach-load-balancers --auto-scaling-group-name my-asg \
```

```
  --load-balancer-names my-lb
```

添加 Elastic Load Balancing 运行状况检查

要将 Elastic Load Balancing 运行状况检查添加到 Auto Scaling 组，请运行以下 update-auto-scaling-group 命令并指定 ELB 作为 --health-check-type 选项的值。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-lb-asg \
  --health-check-type ELB
```

要更新运行状况检查宽限期，请使用 --health-check-grace-period 选项。新实例通常需要时间进行短暂的热身，然后才能通过运行状况检查。如果宽限期没有提供足够的预热时间，则实例可能未准备好提供流量。Amazon EC2 Auto Scaling 可能会将这些实例视为运行状况不佳并替换它们。有关更多信息，请参阅设置自动扩缩组的运行状况检查宽限期 (p. 246)。

以下 update-auto-scaling-group 命令将添加 Elastic Load Balancing 运行状况检查，并指定 300 秒的宽限期。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-lb-asg \
  --health-check-type ELB --health-check-grace-period 300
```

更新可用区

您使用的命令取决于您的负载均衡器是应用程序负载均衡器还是网络负载均衡器，又或者是经典负载均衡器。仅当负载均衡器支持时，才可更新负载均衡器的子网和可用区。有关更多信息，请参阅限制 (p. 279)。

对于具有 Application Load Balancer 或 Network Load Balancer 的 Auto Scaling 组

1. 使用以下 update-auto-scaling-group 命令指定用于 Auto Scaling 组的子网。

   ```
   aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
     --vpc-zone-identifier subnet-41767929 subnet-cb663da2 subnet-8360a9e7
   ```

2. 使用以下 describe-auto-scaling-groups 命令验证新子网中的实例是否准备好接受来自负载均衡器的流量。

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
   ```

3. 使用以下 set-subnets 命令指定用于应用 Application Load Balancer 或 Network Load Balancer 的子网。

   ```
   aws elbv2 set-subnets --load-balancer-arn my-lb-arn \
     --subnets subnet-41767929 subnet-cb663da2 subnet-8360a9e7
   ```

对于 VPC 中具有经典负载均衡器的 Auto Scaling 组

1. 使用以下 update-auto-scaling-group 命令指定用于 Auto Scaling 组的子网。

   ```
   aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
     --vpc-zone-identifier subnet-41767929 subnet-cb663da2
   ```

2. 使用以下 describe-auto-scaling-groups 命令验证新子网中的实例是否准备好接受来自负载均衡器的流量。

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
   ```

3. 使用以下 attach-load-balancer-to-subnets 命令为经典负载均衡器启用新子网。

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-lb \
  --subnets subnet-cb663da2
```

要禁用子网，请运行以下 detach-load-balancer-from-subnets 命令。

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-lb \
  --subnets subnet-8360a9e7
```

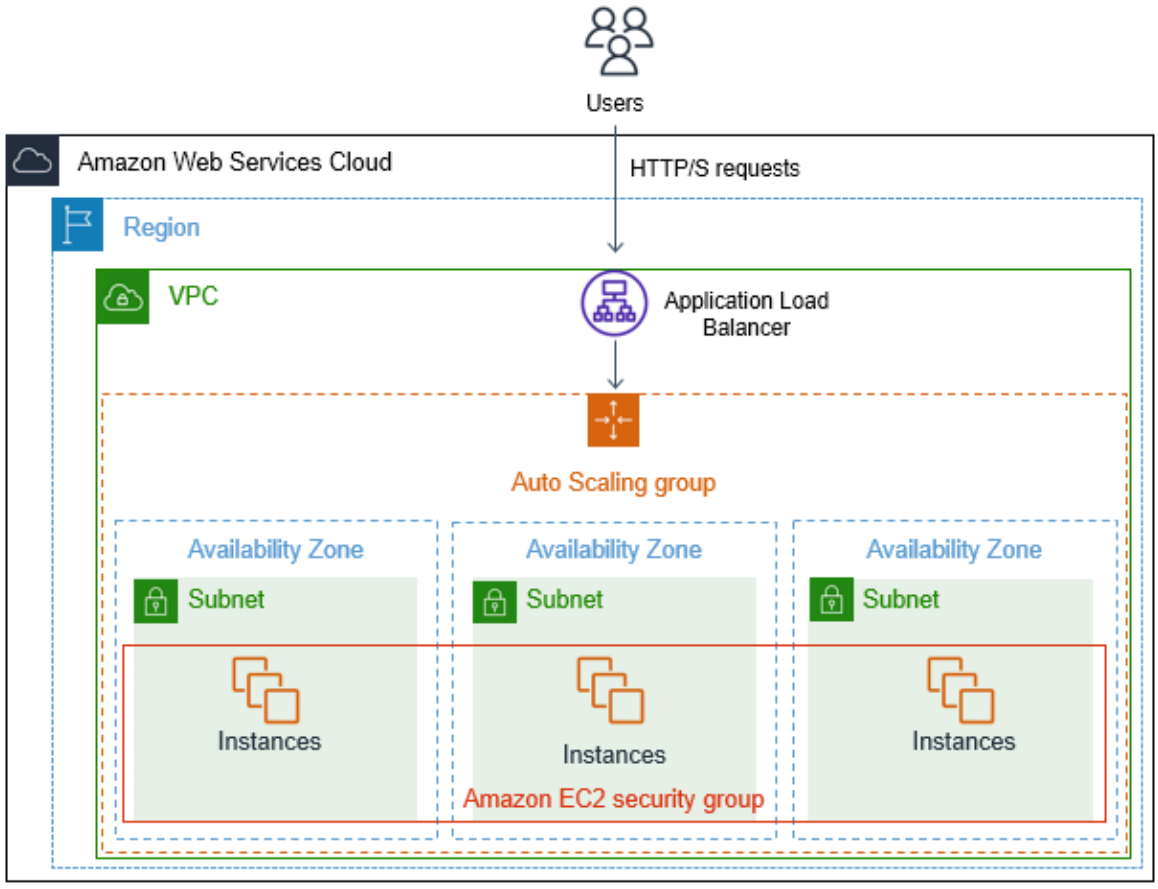
教程：设置具有扩展和负载均衡功能的应用程序

Important

在开始学习本教程之前，我们建议您首先阅读以下介绍性教程：Amazon EC2 Auto Scaling 入门 (p. 13)。

使用 Elastic Load Balancing 负载均衡器注册 Auto Scaling 组可帮助您设置具有负载均衡功能的应用程序。Elastic Load Balancing 与 Amazon EC2 Auto Scaling 配合使用，在运行状况良好的 Amazon EC2 实例之间分配传入流量。这将提高应用程序的可扩展性和可用性。您可以在多个可用区内启用 Elastic Load Balancing 来提高应用程序的容错能力。

在本教程中，我们介绍了在创建 Auto Scaling 组时设置负载均衡的应用程序的基本步骤。完成后，您的架构看起来应当如下图所示：



Elastic Load Balancing 支持三类负载均衡器。我们建议您在本教程中使用 Application Load Balancer。

有关在架构中引入负载均衡器的详细信息，请参阅 使用 Elastic Load Balancing 跨 Auto Scaling 组中的实例分配流量。 (p. 272)。

任务

- 先决条件 (p. 283)
- 步骤 1：设置启动模板或启动配置 (p. 283)
- 步骤 2：创建 Auto Scaling 组。 (p. 285)
- 步骤 3：验证是否已附加您的负载均衡器 (p. 286)
- 步骤 4：后续步骤 (p. 286)
- 第 5 步：清除 (p. 286)

先决条件

- 负载均衡器和目标组。确保针对计划为 Auto Scaling 组使用的负载均衡器选择了相同的可用区。有关更多信息，请参阅 Elastic Load Balancing 用户指南中的 Elastic Load Balancing 入门。
- 启动模板或启动配置的安全组。安全组必须允许从侦听器端口（通常为 HTTP 流量的端口 80）和您希望 Elastic Load Balancing 用于运行状况检查的端口上的负载均衡器进行访问。有关更多信息，请参阅相应文档：
  - Application Load Balancer 用户指南中的目标安全组
  - 《Network Load Balancer 用户指南》中的目标安全组

  如果您的实例将拥有公有 IP 地址，您可以选择允许 SSH 流量连接到这些实例。
- （可选）IAM 角色，授予您的应用程序访问 Amazon 的权限。
- （可选）Amazon Machine Image (AMI)，已定义为 Amazon EC2 实例的源模板。要立即创建一个上述项，请启动一个实例。将 IAM 角色（如果已创建）和所需的任何配置脚本指定为用户数据。连接到实例并对其进行自定义。例如，您可以安装软件和应用程序、复制数据和连接更多的 EBS 卷。测试您的实例上的应用程序以确保实例配置正确。将此更新的配置另存为自定义 AMI。如果您以后不需要该实例，您可以终止它。从该新自定义 AMI 启动的实例包括您在创建 AMI 时设置的自定义项。
- Virtual Private Cloud (VPC)。本教程引用默认 VPC，但您可以使用自己的 VPC。如果使用您自己的 VPC，请确保它拥有映射到您工作时所在区域的每个可用区的子网。您至少必须具有两个公有子网，且这些子网可用于创建负载均衡器。您还必须具有两个私有子网或两个公有子网，以创建 Auto Scaling 组并使用负载均衡器注册。

步骤 1：设置启动模板或启动配置

对本教程使用启动模板或启动配置。

如果您已拥有要使用的启动模板，请使用以下过程选择该启动模板。

> **Note**
>
> 或者，您可以使用启动配置，而不是使用启动模板。有关启动配置说明，请参阅 创建或选择启动配置 (p. 284)。

选择现有启动模板

1. 打开 Amazon EC2 控制台的启动模板页面。
2. 在屏幕顶部的导航栏上，选择在其中创建了负载均衡器的区域。
3. 选择启动模板。
4. 选择 Actions (操作)、Create Auto Scaling group (创建 Auto Scaling 组)。

或者，使用下列过程创建新的启动模板。

创建启动模板

1. 打开 Amazon EC2 控制台的启动模板页面。
2. 在屏幕顶部的导航栏上，选择在其中创建了负载均衡器的区域。
3. 选择 Create launch template (创建启动模板)。
4. 为启动模板的初始版本输入名称并提供描述。
5. 对于 Application and OS Images (Amazon Machine Image) [应用程序和操作系统镜像（Amazon Machine Image）]，选择实例 AMI 的 ID。您可以搜索所有可用的 AMI，也可以从 Recents（最近）或 Quick Start（快速启动）列表中选择一个 AMI。如果您没有看到所需的 AMI，请选择 Browser more AMIs（浏览更多 AMI）以浏览完整的 AMI 目录。
6. 对于 Instance type (实例类型)，选择与您指定的 AMI 兼容的实例硬件配置。
7. （可选）对于 Key pair (login) (密钥对（登录）)，请输入在连接到您的实例时使用的密钥对。
8. 对于 Network settings（网络设置），展开 Advanced network configuration（高级网络配置），然后执行以下操作：
   a. 选择 Add network interface（添加网络接口）以配置主网络接口。
   b. （可选）对于 Auto-assign public IP (自动分配公有 IP)，请保留默认值 Don't include in launch template (不包含在启动模板中)。创建 Auto Scaling 组时，您可以使用已启用公有 IPv4 寻址属性的子网（如默认 VPC 中的默认子网）为 Auto Scaling 组中的实例分配公有 IPv4 地址。或者，如果不需要连接到您的实例，可以选择禁用以防止组中的实例直接从互联网接收流量。在这种情况下，它们将仅从负载均衡器接收流量。
   c. 对于 Security group ID (安全组 ID)，从与负载均衡器相同的 VPC 中为您的实例指定一个安全组。
   d. 对于 Delete on termination (终止时删除)，请选择 Yes (是)。这将在 Auto Scaling 组缩减并终止网络接口附加到的实例时删除网络接口。
9. （可选）要将凭证安全地分配到实例，对于 Advanced details (高级详细信息) 和 IAM instance profile (IAM 实例配置文件)，输入 IAM 角色的 Amazon Resource Name (ARN)。
10. (可选) 要为实例指定用户数据或配置脚本，请将其粘贴到高级详细信息和用户数据。
11. 选择Create launch template（创建启动模板）。
12. 在确认页面上，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

创建或选择启动配置

如果您已拥有要使用的启动配置，请使用以下过程选择该启动配置。

选择现有的启动配置

1. 打开 Amazon EC2 控制台的启动配置页面。
2. 在屏幕顶部的导航栏上，选择在其中创建了负载均衡器的区域。
3. 选择启动配置。
4. 选择操作、创建 Auto Scaling 组。

或者，要创建新的启动配置，请使用以下过程：

创建启动配置

1. 打开 Amazon EC2 控制台的启动配置页面。
2. 在屏幕顶部的导航栏上，选择在其中创建了负载均衡器的区域。
3. 选择创建启动配置，然后为您的启动配置输入名称。
4. 对于 Amazon Machine Image (AMI)，请输入您的实例的 AMI 的 ID 作为搜索条件。
5. 对于实例类型，为您的实例选择硬件配置。

6. 在其他配置下，请注意以下字段：

   a. （可选）为将凭证安全分配到您的 EC2 实例，请为 IAM 实例配置文件选择您的 IAM 角色。有关更多信息，请参阅在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)。

   b. （可选）要为实例指定用户数据或配置脚本，请将其粘贴到高级详细信息和用户数据。

   c. （可选）对于高级详细信息、IP 地址类型，保留默认值。创建 Auto Scaling 组时，您可以使用已启用公有 IP 寻址属性的子网（如默认 VPC 中的默认子网）为 Auto Scaling 组中的实例分配公有 IP 地址。或者，如果不需要连接到您的实例，可以选择 Do not assign a public IP address to any instances (不为任何实例分配公有 IP 地址）以防止组中的实例直接从互联网接收流量。在这种情况下，它们将仅从负载均衡器接收流量。

7. 对于安全组，从与负载均衡器相同的 VPC 中选择现有安全组。如果您保留选中创建新安全组选项，将为运行 Linux 的 Amazon EC2 实例配置默认 SSH 规则。将为运行 Windows 的 Amazon EC2 实例配置默认 RDP 角色。

8. 对于密钥对（登录），请选择密钥对选项下的选项。

   如果您已配置 Amazon EC2 实例密钥对，则可在此处选择它。

   如果您尚未拥有 Amazon EC2 实例密钥对，请选择创建新密钥对并为密钥对指定一个可识别的名称。选择下载密钥对以将密钥对下载到您的计算机。

   Important

   如果需要连接到您的实例，不要选择在没有密钥对的情况下继续。

9. 选中确认复选框，然后选择 Create launch configuration（创建启动配置）。

10. 选中新启动配置名称旁边的复选框，然后选择操作、创建 Auto Scaling 组。

步骤 2：创建 Auto Scaling 组。

创建或选择启动模板或启动配置后，使用以下过程从中断的地方继续操作。

创建 Auto Scaling 组

1. 在 Choose launch template or configuration（选择启动模板或配置）页面上，对于 Auto Scaling group name（Auto Scaling 组名称），输入 Auto Scaling 组的名称。

2. [仅启动模板] 对于 Launch template（启动模板），选择 Auto Scaling 组在扩展时使用启动模板的默认版本、最新版本还是特定版本。

3. 选择 Next（下一步）。

   此时将显示页面 Choose instance launch options（选择实例启动选项），以便选择您希望 Auto Scaling 组使用的 VPC 网络设置，并提供启动按需型实例和竞价型实例的选项（如果您选择了启动模板）。

4. 在 Network（网络）部分中，对于 VPC，选择您用于负载均衡器的 VPC。如果您选择了默认 VPC，则它会自动配置为向您的实例提供 Internet 连接。此 VPC 在区域的每个可用区中均包含一个公有子网。

5. 对于 Availability Zones and subnets（可用区和子网），根据负载均衡器所在的可用区从要包含的每个可用区中选择一个或多个子网。有关更多信息，请参阅 选择 VPC 子网时的注意事项 (p. 302)。

6. [仅限启动模板] 在 Instance type requirements（实例类型要求）部分中，使用默认设置简化此步骤。（请勿覆盖启动模板。）在本教程中，您将仅使用启动模板中指定的实例类型启动按需实例。

7. 选择 Next（下一步）转至 Configure advanced options（配置高级选项）页面。

8. 要将组连接到现有的负载均衡器，请在 Load balancing (负载均衡) 部分中选择 Attach to an existing load balancer (连接到现有负载均衡器)。您可以选择 Choose from your load balancer target groups (从负载均衡器目标组中进行选择) 或 Choose from Classic Load Balancers (从经典负载均衡器中选择)。然后，您可以为创建的 Application Load Balancer 或 Network Load Balancer 选择目标组的名称，或者选择经典负载均衡器的名称。

9. （可选）要使用 Elastic Load Balancing 运行状况检查，对于 Health checks（运行状况检查），请选择 Health check type（运行状况检查类型）下的 ELB。

10. 完成 Auto Scaling 组的配置后，选择 Skip to review（跳过以审核）。
11. 在 Review（审核）页面上，审核 Auto Scaling 组的详细信息。您可以选择 Edit（编辑）进行更改。在完成后，选择 Create Auto Scaling group（创建 Auto Scaling 组）。

在创建附加了负载均衡器的 Auto Scaling 组后，负载均衡器会在新实例联机时自动注册这些实例。此时，您只有一个实例，因此要注册的内容并不多。不过，您可以通过更新组的所需容量来添加其他实例。如需分步指导，请参阅 手动扩展 (p. 123)。

步骤 3：验证是否已附加您的负载均衡器

验证是否已附加您的负载均衡器

1. 从 Amazon EC2 控制台的自动扩缩组页面中，选择您的自动扩缩组旁边的复选框。
2. 在详细信息选项卡上，负载均衡将显示任何附加的负载均衡器目标组或经典负载均衡器。
3. 在 Activity（活动）选项卡上，在Activity history（活动历史记录）中，您可以验证您的实例是否已成功启动。Status（状态）列显示 Auto Scaling 组是否具有已成功启动的实例。如果您的实例无法启动，您可以在 Amazon EC2 Auto Scaling 问题排查 (p. 341) 中找到常见实例启动问题的故障排除思路。
4. 在 Instance management（实例管理）选项卡上的 Instances（实例）下，可以验证您的实例是否准备好接收流量。最初，您的实例处于 `Pending` 状态。在实例准备好接收流量时，其状态为 `InService`。Health status（运行状况）列显示对您的实例执行 Amazon EC2 Auto Scaling 运行状况检查的结果。尽管实例可能标记为运行状况良好，但负载均衡器只会向通过负载均衡器运行状况检查的实例发送流量。
5. 验证您已向负载均衡器注册您的实例。打开 Amazon EC2 控制台的目标组页面。选择您的目标组，然后选择 Targets（目标）选项卡。如果实例的状态为 `initial`，这可能是因为它们仍在注册过程中，或者它们仍在进行运行状况检查。当实例状态为 `healthy` 时，即可供使用。

步骤 4：后续步骤

现在您已完成本教程，您可以了解更多信息：

- 您可以将 Auto Scaling 组配置为使用 Elastic Load Balancing 运行状况检查。如果您启用负载均衡器运行状况检查，并且一个实例未通过运行状况检查，Auto Scaling 组即认为该实例运行状况不正常并进行替换。有关更多信息，请参阅添加 Elastic Load Balancing 运行状况检查 (p. 276)。
- 可以将应用程序扩展到同一区域中的其他可用区，来提高服务中断时的容错能力。有关更多信息，请参阅添加可用区 (p. 277)。
- 您可以将 Auto Scaling 组配置为使用目标跟踪扩展策略。这会在实例需求变化时自动增加或减少实例数量。这将允许该组处理应用程序接收的流量的变化。有关更多信息，请参阅目标跟踪扩缩策略 (p. 133)。

第 5 步：清除

完成为本教程创建的资源后，应考虑清除这些资源，以免产生不必要的费用。

要删除 Auto Scaling 组

1. 打开 Amazon EC2 控制台的自动扩缩组页面。
2. 选中您的 Auto Scaling 组旁边的复选框。
3. 选择 Delete (删除)。
4. 当系统提示进行确认时，键入 **`delete`** 以确认删除指定自动扩缩组，然后选择 Delete（删除）。

   名称列中的加载图标指示 Auto Scaling 组正在被删除。发生删除时，Desired（所需）、Min（最小）和 Max（最大）列显示 Auto Scaling 组具有 `0` 个实例。终止实例并删除组需要几分钟时间。刷新列表以查看当前状态。

如果要保留启动模板，请跳过以下过程。

删除启动模板

1. 打开 Amazon EC2 控制台的启动模板页面。
2. 选择启动模板。
3. 选择 Actions（操作），然后选择 Delete template（删除模板）。
4. 当系统提示进行确认时，键入 **Delete** 以确认删除指定启动模板，然后选择 Delete（删除）。

如果您要保留启动配置，请跳过以下过程。

删除启动配置

1. 打开 Amazon EC2 控制台的启动配置页面。
2. 选择启动配置。
3. 依次选择 Actions（删除）和 Delete launch configuration（删除启动配置）。
4. 当系统提示进行确认时，选择 Delete（删除）。

如果您要保留负载均衡器供将来使用，请跳过以下步骤。

删除您的负载均衡器

1. 打开 Amazon EC2 控制台的负载均衡器页面。
2. 选择负载均衡器，然后依次选择 Actions（操作）和 Delete（删除）。
3. 当系统提示进行确认时，选择 Yes, Delete（是，删除）。

要删除目标组

1. 打开 Amazon EC2 控制台的目标组页面。
2. 选择目标组，然后依次选择 Actions（操作）和 Delete（删除）。
3. 当系统提示进行确认时，选择 Yes, Delete（是，删除）。

使用 EventBridge 处理 Auto Scaling 事件

Amazon EventBridge，以前称为 CloudWatch Events，有助于您设置事件驱动的规则，这些规则可监控资源并启动使用其他 Amazon 服务的目标操作。

Amazon EC2 Auto Scaling 中的事件将近乎实时传输到 EventBridge。您可以建立调用编程操作和通知以响应各种这些事件的 EventBridge 规则。例如，当实例处于启动或终止过程时，您可以调用 Amazon Lambda 函数来执行预配置的任务。

EventBridge 规则的目标可以包括 Amazon Lambda 函数、Amazon SNS 主题、API 目的地、其他 Amazon Web Services 账户中的事件总线等等。有关支持的目标信息，请参阅 Amazon EventBridge 用户指南中的 Amazon EventBridge 目标。

首先使用具有一个 Amazon SNS 主题和一个 EventBridge 规则的示例来创建 EventBridge 规则。然后在用户启动实例刷新后，每当到达某个检查点时，Amazon SNS 都会通过电子邮件通知您。有关更多信息，请参阅为实例刷新事件创建 EventBridge 规则 (p. 297)。

目录

- Amazon EC2 Auto Scaling 事件参考 (p. 288)
- 暖池活动类型和模式 (p. 293)
- 创建 EventBridge 规则 (p. 297)

Amazon EC2 Auto Scaling 事件参考

使用 Amazon EventBridge，您可以创建匹配传入事件的规则并将它们路由到目标进行处理。

示例事件

以下示例演示了 Amazon EC2 Auto Scaling 的事件。事件会尽可能生成。

事件类型

- EC2 实例启动生命周期操作 (p. 288)
- EC2 实例启动成功 (p. 289)
- EC2 实例启动失败 (p. 289)
- EC2 实例终止生命周期操作 (p. 290)
- EC2 实例终止成功 (p. 290)
- EC2 实例终止失败 (p. 291)
- 已达到 EC2 Auto Scaling 实例刷新检查点 (p. 291)
- EC2 Auto Scaling 实例刷新已启动 (p. 292)
- EC2 Auto Scaling 实例刷新成功 (p. 292)
- EC2 Auto Scaling 实例刷新失败 (p. 292)
- EC2 Auto Scaling 实例刷新已取消 (p. 293)

EC2 实例启动生命周期操作

在此示例事件中，由于生命周期挂钩的原因，Amazon EC2 Auto Scaling 已将一个实例移至 `Pending:Wait` 状态。

> **Note**
>
> Amazon EC2 Auto Scaling 支持在生命周期挂钩的事件中提供 `Origin` 和 `Destination`（如果该信息可用）。有关更多信息，请参阅暖池活动类型和模式 (p. 293)。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken": "87654321-4321-4321-4321-210987654321",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
    "NotificationMetadata": "additional-info"
```

```
  }
}
```

EC2 实例启动成功

在此示例事件中，Amazon EC2 Auto Scaling 成功启动了一个实例。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
      "auto-scaling-group-arn",
      "instance-arn"
  ],
  "detail": {
      "StatusCode": "InProgress",
      "Description": "Launching a new EC2 instance: i-12345678",
      "AutoScalingGroupName": "my-auto-scaling-group",
      "ActivityId": "87654321-4321-4321-4321-210987654321",
      "Details": {
          "Availability Zone": "us-west-2b",
          "Subnet ID": "subnet-12345678"
      },
      "RequestId": "12345678-1234-1234-1234-123456789012",
      "StatusMessage": "",
      "EndTime": "yyyy-mm-ddThh:mm:ssZ",
      "EC2InstanceId": "i-1234567890abcdef0",
      "StartTime": "yyyy-mm-ddThh:mm:ssZ",
      "Cause": "description-text"
  }
}
```

EC2 实例启动失败

在此示例事件中，Amazon EC2 Auto Scaling 未能启动一个实例。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Launch Unsuccessful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn",
    "instance-arn"
  ],
  "detail": {
      "StatusCode": "Failed",
      "AutoScalingGroupName": "my-auto-scaling-group",
      "ActivityId": "87654321-4321-4321-4321-210987654321",
      "Details": {
          "Availability Zone": "us-west-2b",
          "Subnet ID": "subnet-12345678"
      },
      "RequestId": "12345678-1234-1234-1234-123456789012",
```

```
      "StatusMessage": "message-text",
      "EndTime": "yyyy-mm-ddThh:mm:ssZ",
      "EC2InstanceId": "i-1234567890abcdef0",
      "StartTime": "yyyy-mm-ddThh:mm:ssZ",
      "Cause": "description-text"
  }
}
```

EC2 实例终止生命周期操作

在此示例事件中，由于生命周期挂钩的原因，Amazon EC2 Auto Scaling 已将一个实例移至 `Terminating:Wait` 状态。

> **Note**
>
> Amazon EC2 Auto Scaling 支持在生命周期挂钩的事件中提供 `Origin` 和 `Destination`（如果该信息可用）。有关更多信息，请参阅暖池活动类型和模式 (p. 293)。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-terminate Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken":"87654321-4321-4321-4321-210987654321",
    "AutoScalingGroupName":"my-asg",
    "LifecycleHookName":"my-lifecycle-hook",
    "EC2InstanceId":"i-1234567890abcdef0",
    "LifecycleTransition":"autoscaling:EC2_INSTANCE_TERMINATING",
    "NotificationMetadata":"additional-info"
  }
}
```

EC2 实例终止成功

在此示例事件中，Amazon EC2 Auto Scaling 成功终止了一个实例。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Terminate Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn",
    "instance-arn"
  ],
  "detail": {
      "StatusCode": "InProgress",
      "Description": "Terminating EC2 instance: i-12345678",
      "AutoScalingGroupName": "my-auto-scaling-group",
      "ActivityId": "87654321-4321-4321-4321-210987654321",
      "Details": {
          "Availability Zone": "us-west-2b",
```

```
          "Subnet ID": "subnet-12345678"
      },
      "RequestId": "12345678-1234-1234-1234-123456789012",
      "StatusMessage": "",
      "EndTime": "yyyy-mm-ddThh:mm:ssZ",
      "EC2InstanceId": "i-1234567890abcdef0",
      "StartTime": "yyyy-mm-ddThh:mm:ssZ",
      "Cause": "description-text"
  }
}
```

EC2 实例终止失败

在此示例事件中，Amazon EC2 Auto Scaling 未能终止一个实例。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Terminate Unsuccessful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn",
    "instance-arn"
  ],
  "detail": {
      "StatusCode": "Failed",
      "AutoScalingGroupName": "my-auto-scaling-group",
      "ActivityId": "87654321-4321-4321-4321-210987654321",
      "Details": {
          "Availability Zone": "us-west-2b",
          "Subnet ID": "subnet-12345678"
      },
      "RequestId": "12345678-1234-1234-1234-123456789012",
      "StatusMessage": "message-text",
      "EndTime": "yyyy-mm-ddThh:mm:ssZ",
      "EC2InstanceId": "i-1234567890abcdef0",
      "StartTime": "yyyy-mm-ddThh:mm:ssZ",
      "Cause": "description-text"
  }
}
```

已达到 EC2 Auto Scaling 实例刷新检查点

在此示例事件中，已替换的实例数量达到了为该检查点定义的百分比阈值。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Checkpoint Reached",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
      "InstanceRefreshId": "ab00cf8f-9126-4f3c-8010-dbb8cad6fb86",
      "AutoScalingGroupName": "my-auto-scaling-group",
```

```
      "CheckpointPercentage": "50",
      "CheckpointDelay": "300"
  }
}
```

EC2 Auto Scaling 实例刷新已启动

在此示例事件中，实例刷新的状态更改为 InProgress。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Started",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
      "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
      "AutoScalingGroupName": "my-auto-scaling-group"
  }
}
```

EC2 Auto Scaling 实例刷新成功

在此示例事件中，实例刷新的状态更改为 Succeeded。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Succeeded",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
      "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
      "AutoScalingGroupName": "my-auto-scaling-group"
  }
}
```

EC2 Auto Scaling 实例刷新失败

在此示例事件中，实例刷新的状态更改为 Failed。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Failed",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
```

```
    "auto-scaling-group-arn"
  ],
  "detail": {
      "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
      "AutoScalingGroupName": "my-auto-scaling-group"
  }
}
```

EC2 Auto Scaling 实例刷新已取消

在此示例事件中，实例刷新的状态更改为 `Cancelled`。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Cancelled",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
      "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
      "AutoScalingGroupName": "my-auto-scaling-group"
  }
}
```

暖池活动类型和模式

> **Note**
>
> Amazon EC2 Auto Scaling 支持 Amazon EventBridge 中的多种预定义模式。这简化了事件模式的创建方式。您可以选择表单上的字段值，然后 EventBridge 将为您生成模式。目前，对于具有暖池的自动扩缩组发出的任何事件，Amazon EC2 Auto Scaling 不支持预定义模式。您必须以 JSON 对象的形式输入模式。这一部分和 为暖池事件创建 EventBridge 规则 (p. 299) 主题演示例了如何使用事件模式来选择事件并将其发送到目标。

将生命周期挂钩添加到自动扩缩组时，事件会在实例变为等待状态时发送到 EventBridge。

生命周期挂钩有两种主要事件类型：

- `EC2 Instance-launch Lifecycle Action`
- `EC2 Instance-terminate Lifecycle Action`

要创建 EventBridge 规则来筛选由具有温池的自动扩缩组发出的特定事件，请包含事件的 `detail` 部分的 `Origin` 和 `Destination` 字段。包含 `Origin` 和 `Destination` 字段的事件仅可由具有暖池的自动扩缩组发出。

它可以是以下 `Origin` 和 `Destination` 值之一：

`EC2 | AutoScalingGroup | WarmPool`

目录

- 示例事件 (p. 294)
- 示例事件模式 (p. 296)

示例事件

这一部分列出了实例变为等待状态时将发送到 EventBridge 的事件。尽最大努力发出事件。

事件示例

- 示例 1：Amazon EC2 Auto Scaling 将一个新实例添加到暖池 (p. 294)
- 示例 2：Amazon EC2 Auto Scaling 将一个实例添加到自动扩缩组 (p. 294)
- 示例 3：Amazon EC2 Auto Scaling 将一个新实例添加到自动扩缩组 (p. 295)
- 示例 4：Amazon EC2 Auto Scaling 将一个实例退回暖池 (p. 295)

示例 1：Amazon EC2 Auto Scaling 将一个新实例添加到暖池

在此示例事件中，新实例的状态在其被添加到暖池时更改为 `Warmed:Pending:Wait`。其原因在于横向扩展事件的生命周期挂钩。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2021-01-13T00:12:37.214Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:042cba90-
ad2f-431c-9b4d-6d9055bcc9fb:autoScalingGroupName/my-asg"
  ],
  "detail": {
    "LifecycleActionToken": "71514b9d-6a40-4b26-8523-05e7eEXAMPLE",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-launch-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
    "NotificationMetadata": "additional-info",
    "Origin": "EC2",
    "Destination": "WarmPool"
  }
}
```

示例 2：Amazon EC2 Auto Scaling 将一个实例添加到自动扩缩组

在此示例事件中，暖池中实例的状态在其被添加到自动扩缩组时更改为 `Pending:Wait`。其原因在于横向扩展事件的生命周期挂钩。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2021-01-19T00:35:52.359Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:042cba90-
ad2f-431c-9b4d-6d9055bcc9fb:autoScalingGroupName/my-asg"
  ],
  "detail": {
    "LifecycleActionToken": "19cc4d4a-e450-4d1c-b448-0de67EXAMPLE",
    "AutoScalingGroupName": "my-asg",
```

```
    "LifecycleHookName": "my-launch-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
    "NotificationMetadata": "additional-info",
    "Origin": "WarmPool",
    "Destination": "AutoScalingGroup"
  }
}
```

示例 3：Amazon EC2 Auto Scaling 将一个新实例添加到自动扩缩组

在此示例事件中，新实例（而非暖池中的实例）的状态在其被添加到自动扩缩组时更改为 `Pending:Wait`。其原因在于横向扩展事件的生命周期挂钩。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2021-02-01T17:18:06.082Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:042cba90-
ad2f-431c-9b4d-6d9055bcc9fb:autoScalingGroupName/my-asg"
  ],
  "detail": {
    "LifecycleActionToken": "87654321-4321-4321-4321-21098EXAMPLE",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-launch-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
    "NotificationMetadata": "additional-info",
    "Origin": "EC2",
    "Destination": "AutoScalingGroup"
  }
}
```

示例 4：Amazon EC2 Auto Scaling 将一个实例退回暖池

在此示例事件中，实例的状态在其被退回到暖池时更改为 `Warmed:Pending:Wait`。其原因在于横向缩减事件的生命周期挂钩。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-terminate Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2022-03-28T00:12:37.214Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:042cba90-
ad2f-431c-9b4d-6d9055bcc9fb:autoScalingGroupName/my-asg"
  ],
  "detail": {
    "LifecycleActionToken": "42694b3d-4b70-6a62-8523-09a1eEXAMPLE",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-termination-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
    "NotificationMetadata": "additional-info",
```

```
    "Origin": "AutoScalingGroup",
    "Destination": "WarmPool"
  }
}
```

示例事件模式

前一节提供 Amazon EC2 Auto Scaling 发出的示例事件。

EventBridge 事件模式与它们匹配的事件具有相同的结构。模式引用了您要匹配的字段，并提供您所查找的值。

事件中的下列字段构成规则中定义的事件模式以调用操作：

`"source": "aws.autoscaling"`

识别来自 Amazon EC2 Auto Scaling 的事件。

`"detail-type": "EC2 Instance-launch Lifecycle Action"`

识别事件类型。

`"Origin": "EC2"`

标识实例的来源。

`"Destination": "WarmPool"`

标识实例的目标位置。

使用以下示例事件模式捕获与进入暖池的实例相关联的所有事件。

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "Origin": [ "EC2" ],
      "Destination": [ "WarmPool" ]
   }
}
```

使用以下示例事件模式捕获与因横向扩展事件而离开暖池的实例有关的所有事件。

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "Origin": [ "WarmPool" ],
      "Destination": [ "AutoScalingGroup" ]
   }
}
```

使用以下示例事件模式捕获与直接启动到 Auto Scaling 组的实例相关联的所有事件。

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "Origin": [ "EC2" ],
      "Destination": [ "AutoScalingGroup" ]
   }
```

```
}
```

使用以下示例事件模式捕获与在横向缩减时退回暖池的实例有关的所有事件。

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-terminate Lifecycle Action" ],
  "detail": {
      "Origin": [ "AutoScalingGroup" ],
      "Destination": [ "WarmPool" ]
   }
}
```

使用以下示例事件模式捕获与 `EC2 Instance-launch Lifecycle Action` 关联的所有相关事件，无论起点或目的地如何。

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ]
}
```

创建 EventBridge 规则

当 Amazon EC2 Auto Scaling 发送事件时，事件通知将作为 JSON 文件发送到 Amazon EventBridge。当事件模式与规则匹配时，您可以编写 EventBridge 规则并自动执行要采取的操作。如果 EventBridge 检测到与规则中定义的模式匹配的事件模式，则 EventBridge 调用规则中指定的一个或多个目标。

您可以使用这一部分的示例过程作为起点。

以下文档也可能会非常有用：

- 要在实例启动时或使用 Lambda 函数终止实例之前对实例执行自定义操作，请参阅教程：配置调用 Lambda 函数的生命周期钩子 (p. 207)。
- 要在 CloudTrail 记录 API 调用时调用 Lambda 函数，请参阅《Amazon EventBridge 用户指南》中的教程：使用 EventBridge 记录 Amazon API 调用。
- 要详细了解如何创建事件规则，请参阅《Amazon EventBridge 用户指南》中的创建对事件作出反应的 Amazon EventBridge 规则。

主题

- 为实例刷新事件创建 EventBridge 规则 (p. 297)
- 为暖池事件创建 EventBridge 规则 (p. 299)

为实例刷新事件创建 EventBridge 规则

以下示例会创建一个将发送电子邮件通知的 EventBridge 规则。每次在实例刷新期间到达某个检查点，自动扩缩组发出事件时，系统都会执行此操作。包括使用 Amazon SNS 设置电子邮件通知的过程。要使用 Amazon SNS 发送电子邮件通知，必须先创建一个主题，然后用您的电子邮件地址订阅该主题。

有关实例刷新功能的更多信息，请参阅基于实例刷新替换 Auto Scaling 实例 (p. 95)。

创建 Amazon SNS 主题

SNS 主题是一个逻辑接入点，即 Auto Scaling 组用来发送通知的通信通道。您可通过为主题指定名称来创建主题。

主题名称必须满足以下要求：

- 使用 1 到 256 个字符
- 包含大写和小写 ASCII 字母、数字、下划线或连字符

有关更多信息，请参阅 Amazon Simple Notification Service 开发人员指南中的创建 Amazon SNS 主题。

订阅 Amazon SNS 主题

要接收您的 Auto Scaling 组发送到该主题的通知，必须让一个终端节点订阅该主题。在此过程中，为端点指定要用来接收来自 Amazon EC2 Auto Scaling 的通知的电子邮件地址。

有关更多信息，请参阅 Amazon Simple Notification Service 开发人员指南中的订阅 Amazon SNS 主题。

确认您的 Amazon SNS 订阅

Amazon SNS 向在上一步骤中指定的电子邮件地址发送确认电子邮件。

确保打开来自 Amazon 通知的电子邮件并选择链接以确认订阅，然后再继续执行下一步。

您将收到来自 Amazon 的确认消息。Amazon SNS 现已配置为接收通知并以电子邮件形式将通知发送到指定的电子邮件地址。

将事件路由到您的 Amazon SNS 主题

创建匹配选定事件的规则，并将它们路由到您的 Amazon SNS 主题，以通知订阅的电子邮件地址。

创建向您的 Amazon SNS 主题发送通知的规则

1. 打开位于 https://console.aws.amazon.com/events/ 的 Amazon EventBridge 控制台。
2. 在导航窗格中，选择 Rules (规则)。
3. 选择 Create rule (创建规则)。
4. 对于定义规则详细信息，请执行以下操作：

   a. 输入规则的 Name (名称) 和“Description (描述)”（可选）。

      规则不能与同一区域中的另一个规则和同一事件总线上的名称相同。

   b. 对于 Event bus（事件总线），选择 default（默认）。当您账户中的某个 Amazon 服务生成一个事件时，它始终会发送到您账户的默认事件总线。

   c. 对于 Rule type（规则类型），选择 Rule with an event pattern（具有事件模式的规则）。

   d. 选择 Next（下一步）。

5. 对于 Build event pattern（构建事件模式），执行以下操作：

   a. 对于 Event source（事件源），选择 Amazon 事件或 EventBridge 合作伙伴事件。

   b. 对于 Event pattern（事件模式），执行以下操作：

      i. 对于事件源，选择 Amazon Web Services。

      ii. 对于 Amazon Web Service，选择 Auto Scaling。

      iii. 对于事件类型，选择实例刷新。

      iv. 预设情况下，该规则与任何实例刷新事件匹配。要创建将在实例刷新期间达到某个检查点时通知您的规则，请选择 Specific instance event(s)（特定实例事件），然后选择 EC2 Auto Scaling Instance Refresh Checkpoint Reached（已达到 EC2 Auto Scaling 实例刷新检查点）。

      v. 默认情况下，该规则与区域中任何 Auto Scaling 组匹配。若要使该规则与特定 Auto Scaling 组匹配，请选择 特定组名称并选择一个或多个 Auto Scaling 组。

vi. 选择 Next（下一步）。

6. 对于 Select target(s)（选择目标），请执行以下操作：
   a. 对于 Target types（目标类型），选择 Amazon Web Service。
   b. 对于 Select a target（选择一个目标），选择 SNS topic（SNS 主题）。
   c. 对于 Topic（主题），选择您的 Amazon SNS 主题。
   d. （可选）在 Additional settings（其他设置）下，您可以选择配置其他设置。有关更多信息，请参阅《Amazon EventBridge 用户指南》中的创建对事件作出反应的 Amazon EventBridge 规则（步骤 16）。
   e. 选择 Next（下一步）。
7. （可选）对于 Tags（标签），您可以选择向规则分配一个或多个标签，然后选择 Next（下一步）。
8. 对于 Review and create（检查并创建），检查规则的详细信息并根据需要对其进行修改。然后选择 Create rule（创建规则）。

为暖池事件创建 EventBridge 规则

以下示例创建了一个用来调用编程操作的 EventBridge 规则。每当您的自动扩缩组在有新实例被添加到暖池时发出事件时，都会执行此操作。

在创建规则之前，请创建您希望该规则用作目标的 Amazon Lambda 函数。您必须将此函数指定为该规则的目标。以下过程仅提供了创建将在新实例进入暖池时作出反应的 EventBridge 规则的步骤。有关演示如何创建可在传入的事件与规则匹配时进行调用的简单 Lambda 函数的入门教程，请参阅教程：配置调用 Lambda 函数的生命周期钩子 (p. 207)。

有关创建和使用暖池的更多信息，请参阅 Amazon EC2 Auto Scaling 的暖池 (p. 212)。

创建会调用 Lambda 函数的事件规则

1. 打开位于 https://console.aws.amazon.com/events/ 的 Amazon EventBridge 控制台。
2. 在导航窗格中，选择 Rules (规则)。
3. 选择 Create rule (创建规则)。
4. 对于定义规则详细信息，请执行以下操作：
   a. 输入规则的 Name (名称) 和“Description (描述)”（可选）。

      规则不能与同一区域中的另一个规则和同一事件总线上的名称相同。
   b. 对于 Event bus（事件总线），选择 default（默认）。当您账户中的某个 Amazon Web Service生成一个事件时，它始终会发送到您账户的默认事件总线。
   c. 对于 Rule type（规则类型），选择 Rule with an event pattern（具有事件模式的规则）。
   d. 选择 Next（下一步）。
5. 对于 Build event pattern（构建事件模式），执行以下操作：
   a. 对于 Event source（事件源），选择 Amazon 事件或 EventBridge 合作伙伴事件。
   b. 对于 Event pattern（事件模式），选择 Custom pattern (JSON editor) [自定义模式（JSON 编辑器）]，并将以下模式粘贴到 Event pattern（事件模式）框中，将其中的*##*文本替换为您的自动扩缩组的名称。

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
      "AutoScalingGroupName": [ "my-asg" ],
      "Origin": [ "EC2" ],
      "Destination": [ "WarmPool" ]
```

```
    }
}
```

要创建与其他事件匹配的规则，请修改事件模式。有关更多信息，请参阅示例事件模式 (p. 296)。

c. 选择 Next（下一步）。

6. 对于 Select target(s)（选择目标），请执行以下操作：

   a. 对于 Target types（目标类型），选择 Amazon Web Service。
   b. 对于 Select a target（选择目标），选择 Lambda function（Lambda 函数）。
   c. 然后，对于 Function（函数），选择要将事件发送到的函数。
   d. （可选）对于 Configure version/alias（配置版本/别名），输入目标 Lambda 函数的版本和别名设置。
   e. （可选）对于 Additional settings（其他设置），视应用程序的情况输入任何其他设置。有关更多信息，请参阅《Amazon EventBridge 用户指南》中的创建对事件作出反应的 Amazon EventBridge 规则（步骤 16）。
   f. 选择 Next（下一步）。

7. （可选）对于 Tags（标签），您可以选择向规则分配一个或多个标签，然后选择 Next（下一步）。
8. 对于 Review and create（检查并创建），检查规则的详细信息并根据需要对其进行修改。然后选择 Create rule（创建规则）。

使用 Amazon VPC 为 Auto Scaling 实例提供网络连接

我们即将停用 EC2-Classic。我们建议您从 EC2-Classic 迁移到 VPC。有关更多信息，请参阅博客文章 EC2-Classic Networking is Retiring - Here's How to Prepare。

Amazon Virtual Private Cloud (Amazon VPC) 允许您在 Amazon Web Services 云 的私有、隔离部分定义虚拟网络环境。您的虚拟联网环境完全由您控制。

在 Virtual Private Cloud (VPC) 内，您可启动 Amazon 资源，如 Auto Scaling 组。自动扩缩组在 VPC 中的工作原理与它在 EC2-Classic 中的工作原理基本相同，支持同样的功能集。

Amazon VPC 中的子网是可用区的一个部分，由 VPC 的 IP 地址范围段定义。您可以根据您的安全和运行需求，使用子网对您的实例进行分组。子网完全位于其最初创建时所在的可用区内。在子网中启动 Auto Scaling 实例。

要支持 Internet 与您的子网中的实例之间的通信，您必须创建一个 Internet 网关并将其附加到您的 VPC。通过使用互联网网关，您的子网中的资源可以通过 Amazon EC2 网络边界连接到互联网。如果一个子网的流量被路由到 Internet 网关，这个子网便是公有子网。如果子网的流量不路由到 Internet 网关，则子网称为私有子网。请将公有子网用于必须连接到 Internet 的资源，并将私有子网用于不需要连接到 Internet 的资源。有关向 VPC 中的实例授予互联网访问的更多信息，请参阅 Amazon VPC 用户指南中的访问互联网。

目录

- EC2-Classic (p. 301)
- 默认 VPC (p. 301)
- 非默认 VPC (p. 302)
- 选择 VPC 子网时的注意事项 (p. 302)
- VPC 中的 IP 寻址 (p. 302)
- VPC 中的网络接口 (p. 302)

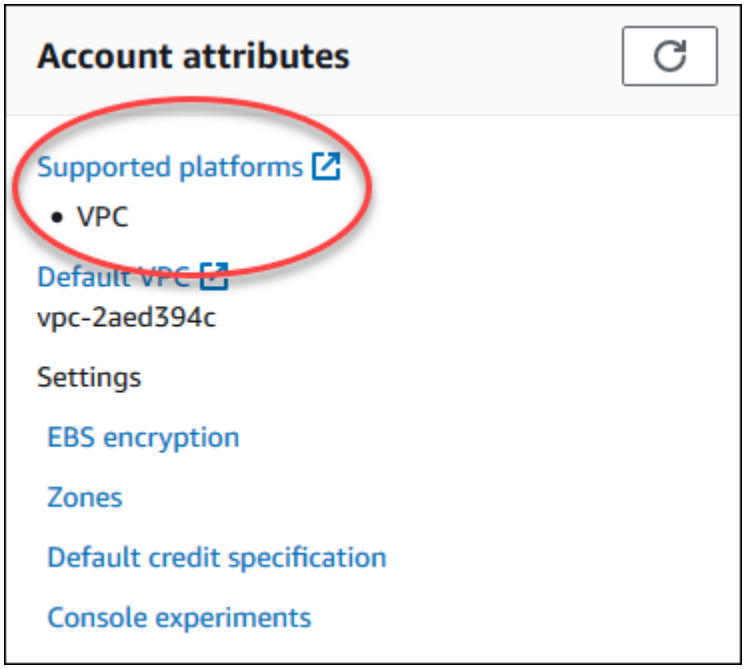
- 实例部署租期 (p. 303)
- Amazon Outposts (p. 303)
- 用于了解 VPC 的更多资源 (p. 303)

EC2-Classic

如果您在 2013 年 12 月 4 日之前创建了 Amazon Web Services 账户，您可以在某些区域中选择 Amazon VPC 和 EC2-Classic。如果您拥有以下较旧账户之一，您可能在某些区域的 EC2-Classic 中具有 Auto Scaling 组，而不是 Amazon VPC。

要了解有关从 EC2 Classic 迁移到 VPC 的信息，请参阅博客文章 EC2-Classic 联网正在停用 - 下面介绍如何准备。有关 EC2-Classic 和 VPC 中实例之间区别的信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 EC2-Classic。

要确定您使用的任何 Amazon Web Services 区域 是否仍在使用 EC2-Classic，请打开 Amazon EC2 控制台。如果 Supported platforms（支持的平台）仅指示 VPC（如下例所示），则说明您在当前 Amazon Web Services 区域 中的 Amazon Web Services 账户 使用 VPC 平台并使用默认 VPC。默认 VPC 的名称显示在支持的平台下方。



Tip

在包含默认 VPC 的区域中创建的任何自动扩缩组都可以在默认 VPC 或非默认 VPC 中启动实例，但不能在 EC2-Classic 中启动实例。

默认 VPC

如果您在 2013 年 12 月 4 日之后创建了 Amazon Web Services 账户，或正在新的 Amazon Web Services 区域 中创建 Auto Scaling 组，我们会为您创建默认 VPC。默认 VPC 随每个可用区中的默认子网提供。如果您有默认 VPC，则在默认情况下，将在默认 VPC 中创建 Auto Scaling 组。

您可以通过 Amazon VPC 控制台的您的 VPC 页面查看您的 VPC。

有关默认 VPC 的更多信息，请参阅《Amazon VPC 用户指南》中的默认 VPC 和默认子网。

非默认 VPC

您可以打开 Amazon Web Services Management Console中的 VPC 控制面板页面，然后选择 Create VPC（创建 VPC）以创建其他 VPC。

有关更多信息，请参阅《Amazon VPC 用户指南》。

> Note
>
> VPC 跨越其 Amazon Web Services 区域中的所有可用区。向 VPC 添加子网时，请选择多个可用区以确保在这些子网中托管的资源具有高可用性。可用区是 Amazon Web Services 区域 中一个或多个具有冗余电源、网络和连接的离散数据中心。可用区可为生产级应用程序提供高可用性、容错能力和可扩展性。

选择 VPC 子网时的注意事项

在为自动扩缩组选择 VPC 子网时，请注意以下事项：

- 如果您要将 Elastic Load Balancing 负载均衡器附加到 Auto Scaling 组，则可以在公有或私有子网中启动实例。但是，只能在公有子网中创建负载均衡器。
- 如果您直接通过 SSH 访问 Auto Scaling 实例，则只能在公有子网中启动实例。
- 如果您正在使用 Amazon Systems Manager 会话管理器访问非入口 Auto Scaling 实例，可以在公有或私有子网中启动实例。
- 如果您使用的是私有子网，则可以使用公有 NAT 网关允许 Auto Scaling 实例访问互联网。
- 预设情况下，默认 VPC 中的默认子网为公有子网。

VPC 中的 IP 寻址

在 VPC 中启动 Auto Scaling 实例时，您的实例将从分配启动实例的子网 CIDR 范围自动分配私有 IP 地址。这样，这些实例就能够与 VPC 中的其他实例通信。

您可以配置启动模板或启动配置以为您的实例分配公有 IPv4 地址。通过将公有 IP 地址分配给实例，这些实例可以与互联网或其他 Amazon 服务进行通信。

在配置为自动分配 IPv6 地址的子网中启动实例时，这些实例会同时收到 IPv4 和 IPv6 地址。否则，这些实例仅接收 IPv4 地址。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 IPv6 地址。

有关为您的 VPC 或子网指定 CIDR 范围的更多信息，请参阅 Amazon VPC 用户指南。

当您使用指定了其他网络接口的启动模板时，Amazon EC2 Auto Scaling 可以在实例启动时自动分配其他私有 IP 地址。每个网络接口都将从启动实例的子网的 CIDR 范围中获得单个私有 IP 地址。在这种情况下，系统将不再能够为主网络接口自动分配公有 IPv4 地址。除非您将可用的弹性 IP 地址与 Auto Scaling 实例关联，否则将无法通过公有 IPv4 地址连接到您的实例。

VPC 中的网络接口

VPC 中的每个实例具有一个默认网络接口（主网络接口）。您无法从实例断开主网络接口。您可以创建其他网络接口并将其挂载至您的 VPC 中的任何实例。您可以挂载的网络接口数因实例类型而有所差异。

使用启动模板启动实例时，您可以指定其他网络接口。但是，启动具有多个网络接口的 Auto Scaling 实例会自动在与实例相同的子网中创建每个接口。这是因为 Amazon EC2 Auto Scaling 会忽略启动模板中定义的子网，而不是 Auto Scaling 组中指定的子网。有关更多信息，请参阅为自动扩缩组创建启动模板。

如果创建来自同一子网的两个或多个网络接口或将其连接到一个实例，可能会遇到非对称路由等联网问题，尤其是使用非 Amazon Linux 变体的实例。如果需要此类配置，则必须在操作系统中配置辅助网络接口。有关示例，请参阅 Amazon 知识中心中的如何在我的 Ubuntu EC2 实例中使用我的辅助网络接口？。

实例部署租期

预设情况下，VPC 中的所有实例将作为共享租期实例运行。Amazon EC2 Auto Scaling 还支持专用实例和专用主机。但是，对专用主机的支持仅适用于使用启动模板的 Auto Scaling 组。有关更多信息，请参阅使用启动配置来配置实例租赁 (p. 46)。

Amazon Outposts

Amazon Outposts 可将 Amazon VPC 从 Amazon 区域扩展到 Outpost，其中包含可在该区域访问的 VPC 组件，包括互联网网关、虚拟专用网关、Amazon VPC 中转网关和 VPC 端点。Outpost 位于该区域内的一个可用区中，是该可用区的延伸，让您可以用来实现弹性。

有关更多信息，请参阅 Amazon Outposts 用户指南。

用于了解 VPC 的更多资源

要了解有关 VPC 和子网的更多信息，请参阅以下主题。

- VPC 中的私有子网
  - 带有公有和私有子网的 VPC (NAT)
  - NAT 网关
- VPC 中的公有子网
  - 带单个公有子网的 VPC
- Application Load Balancer 的子网
  - 您的负载均衡器的子网
- 一般 VPC 信息
  - Amazon VPC User Guide
  - VPC 对等
  - 弹性网络接口
  - 使用 VPC 终端节点建立私有连接 (p. 339)
  - 从 EC2-Classic 迁移到 VPC

Amazon EC2 Auto Scaling 中的安全性

Amazon 的云安全性的优先级最高。作为 Amazon 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是Amazon和您的共同责任。责任共担模型将其描述为云的安全性和云中的安全性：

- 云的安全性 – Amazon 负责保护在 Amazon 云中运行 Amazon 服务的基础设施。Amazon 还向您提供可安全使用的服务。作为 Amazon 合规性计划的一部分，第三方审计人员将定期测试和验证安全措施的有效性。要了解适用于 Amazon EC2 Auto Scaling 的合规性计划，请参阅 Amazon合规性计划范围内的服务。
- 云中的安全性 - 您的责任由您使用的 Amazon 服务决定。您还需要对其它因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

该文档帮助您了解如何在使用 Amazon EC2 Auto Scaling 时应用责任共担模式。以下主题说明如何配置 Amazon EC2 Auto Scaling 以实现您的安全性和合规性目标。您还会了解如何使用其他 Amazon 服务以帮助您监控和保护 Amazon EC2 Auto Scaling 资源。

主题

- Amazon EC2 Auto Scaling 和数据保护 (p. 304)
- 适用于 Amazon EC2 Auto Scaling 的 Identity and Access Management (p. 305)
- Amazon EC2 Auto Scaling 的合规性验证 (p. 337)
- Amazon EC2 Auto Scaling 中的恢复功能 (p. 338)
- Amazon EC2 Auto Scaling 中的基础设施安全性 (p. 338)
- Amazon EC2 Auto Scaling 和接口 VPC 终端节点 (p. 339)

Amazon EC2 Auto Scaling 和数据保护

Amazon 责任共担模式适用于 Amazon EC2 Auto Scaling 中的数据保护。如该模式中所述，Amazon 负责保护运行所有 Amazon Web Services 云 的全球基础设施。您负责维护对托管在此基础设施上的内容的控制。此内容包括您所使用的 Amazon Web Services 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。

出于数据保护目的，我们建议您保护 Amazon Web Services 账户 凭证并使用 Amazon IAM Identity Center (successor to Amazon Single Sign-On) 或 Amazon Identity and Access Management (IAM) 设置单个用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护您的数据：

- 对每个账户使用 Multi-Factor Authentication (MFA)。
- 使用 SSL/TLS 与 Amazon 资源进行通信。建议使用 TLS 1.2 或更高版本。
- 使用 Amazon CloudTrail 设置 API 和用户活动日志记录。
- 使用 Amazon 加密解决方案以及 Amazon Web Services 中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果在通过命令行界面或 API 访问 Amazon 时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准 (FIPS) 第 140-2 版》。

我们强烈建议您切勿将机密信息或敏感信息（例如您客户的电子邮件地址）放入标签或自由格式字段 [例如 Name（名称）字段]。这包括使用控制台、API、Amazon CLI 或 Amazon 开发工具包处理 Amazon EC2 Auto Scaling 或其他 Amazon Web Services 时。您在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，我们强烈建议您不要在 URL 中包含凭证信息来验证您对该服务器的请求。

使用 Amazon KMS keys 对 Amazon EBS 卷进行加密

您可以将 Auto Scaling 组配置为使用 Amazon KMS keys 对存储在云中的 Amazon EBS 卷数据进行加密。Amazon EC2 Auto Scaling 支持 Amazon 托管和客户托管密钥来加密您的数据。请注意，当您使用启动配置时，指定客户托管密钥的 `KmsKeyId` 选项不可用。要指定您的客户托管密钥，请改用启动模板。有关更多信息，请参阅为 Auto Scaling 组创建启动模板 (p. 20)。

在设置启动模板或启动配置之前，您还可以在 EBS 支持的 AMI 中配置客户托管密钥，或在预设情况下使用加密来强制对您创建的新 EBS 卷和快照副本进行加密。

有关如何在使用客户托管密钥进行加密时设置需要启动 Auto Scaling 实例所需的密钥策略的信息，请参阅与加密卷结合使用时必需的 Amazon KMS 密钥策略 (p. 333)。有关如何创建、存储和管理 Amazon KMS 加密密钥的信息，请参阅什么是 Amazon 密钥管理服务？

相关主题

- 适用于 Linux 实例的 Amazon EC2 用户指南中的 Amazon EC2 中的数据保护
- 适用于 Linux 实例的 Amazon EC2 用户指南中的将加密与 EBS 支持的 AMI 结合使用
- 适用于 Linux 实例的 Amazon EC2 用户指南中的默认加密

适用于 Amazon EC2 Auto Scaling 的 Identity and Access Management

Amazon Identity and Access Management (IAM) 是一项 Amazon Web Service，可以帮助管理员安全地控制对 Amazon 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）和授权（具有权限）使用 Amazon EC2 Auto Scaling 资源。IAM 是一项无需额外费用即可使用的 Amazon Web Service。

要使用 Amazon EC2 Auto Scaling，您需要一个 Amazon Web Services 账户 以及用于登录账户的安全凭证。有关更多信息，请参阅《Amazon 一般参考》中的 Amazon 安全凭证。

有关完整的 IAM 文档，请参阅 IAM 用户指南。

访问控制

您可以使用有效的凭证来对自己的请求进行身份验证，但您还必须拥有权限才能创建或访问 Amazon EC2 Auto Scaling 资源。例如，您必须有权创建 Auto Scaling 组、创建启动配置等。

以下部分提供了详细信息来说明 IAM 管理员如何使用 IAM 控制哪些用户可执行 Amazon EC2 Auto Scaling 操作，从而对您的 Amazon EC2 Auto Scaling 资源进行保护。

我们建议您先阅读 Amazon EC2 主题。请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的适用于 Amazon EC2 的 Identity and Access Management。阅读本节中的主题后，您应该很好地了解 Amazon EC2 提供的访问控制权限，以及如何适应您的 Amazon EC2 Auto Scaling 资源权限需求。

主题

- Amazon EC2 Auto Scaling 如何与 IAM 结合使用 (p. 306)
- Amazon EC2 Auto Scaling 的 Amazon 托管策略 (p. 312)
- Amazon EC2 Auto Scaling 的服务相关角色 (p. 314)
- Amazon EC2 Auto Scaling 基于身份的策略示例 (p. 319)
- 防止跨服务混淆代理 (p. 327)
- 启动模板支持 (p. 328)
- 在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)
- 与加密卷结合使用时必需的 Amazon KMS 密钥策略 (p. 333)

Amazon EC2 Auto Scaling 如何与 IAM 结合使用

在使用 IAM 来管理对 Amazon EC2 Auto Scaling 的访问权限之前，请首先了解哪些 IAM 功能可与 Amazon EC2 Auto Scaling 结合使用。

可以与 Amazon EC2 Auto Scaling 结合使用的 IAM 功能

IAM 功能	Amazon EC2 Auto Scaling 支持
基于身份的策略 (p. 306)	是
基于资源的策略 (p. 307)	否
策略操作 (p. 307)	是
策略资源 (p. 308)	是
策略条件键（特定于服务） (p. 309)	是
ACL (p. 310)	否
ABAC（策略中的标签） (p. 310)	部分
临时凭证 (p. 311)	是
服务角色 (p. 311)	是
服务相关角色 (p. 312)	是

要大致了解 Amazon EC2 Auto Scaling 和其他 Amazon Web Services 服务如何与大多数 IAM 功能结合使用，请参阅《IAM 用户指南》中的与 IAM 一起使用的 Amazon Web Services 服务。

Amazon EC2 Auto Scaling 基于身份的策略

支持基于身份的策略	是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的创建 IAM policy。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

Amazon EC2 Auto Scaling 基于身份的策略示例

要查看 Amazon EC2 Auto Scaling 基于身份的策略示例，请参阅 Amazon EC2 Auto Scaling 基于身份的策略示例 (p. 319)。

Amazon EC2 Auto Scaling 内基于资源的策略

支持基于资源的策略。	否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service（Amazon S3）存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。主体可以包括账户、用户、角色、联合身份用户或 Amazon Web Services。

要启用跨账户存取，您可以将整个账户或其它账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当主体和资源处于不同的Amazon Web Services 账户中时，则信任账户中的 IAM 管理员还必须授予主体实体（用户或角色）对资源的访问权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。

Amazon EC2 Auto Scaling 的策略操作

支持策略操作	是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么 条件 下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 Amazon API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行相关操作的权限。

如查看 Amazon EC2 Auto Scaling 操作的列表，请参阅《服务授权参考》中 Amazon EC2 Auto Scaling 定义的操作。

Amazon EC2 Auto Scaling 中的策略操作在操作前面使用以下前缀：

```
autoscaling
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      "autoscaling:action1",
      "autoscaling:action2"
         ]
```

您还可以使用通配符（*）指定多项操作。例如，要指定以单词 `Describe` 开头的所有操作，包括以下操作：

```
"Action": "autoscaling:Describe*"
```

要查看 Amazon EC2 Auto Scaling 基于身份的策略示例，请参阅 Amazon EC2 Auto Scaling 基于身份的策略示例 (p. 319)。

Amazon EC2 Auto Scaling 策略资源

支持策略资源	是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么条件 下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 `Resource` 或 `NotResource` 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符（*）指示语句应用于所有资源。

```
"Resource": "*"
```

您可以使用 ARN 来标识 IAM policy 应用到的自动扩缩组和启动配置。

Auto Scaling 组具有以下 ARN。

```
"Resource": "arn:aws:autoscaling:region:account-
id:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
```

具有以下 ARN 的启动配置。

```
"Resource": "arn:aws:autoscaling:region:account-
id:launchConfiguration:uuid:launchConfigurationName/lc-name"
```

要使用 `CreateAutoScalingGroup` 操作指定自动扩缩组，您必须将 UUID 替换为如下例中所示的通配符（*）。

```
"Resource": "arn:aws:autoscaling:region:account-
id:autoScalingGroup:*:autoScalingGroupName/asg-name"
```

要使用 `CreateLaunchConfiguration` 操作指定启动配置，您必须将 UUID 替换为如下例中所示的通配符（*）。

```
"Resource": "arn:aws:autoscaling:region:account-
id:launchConfiguration:*:launchConfigurationName/lc-name"
```

有关 Amazon EC2 Auto Scaling 资源类型及其 ARN 的更多信息，请参阅《服务授权参考》中的 Amazon EC2 Auto Scaling 定义的资源。要了解您可以用来指定每个资源的 ARN 的操作，请参阅 Amazon EC2 Auto Scaling 定义的操作。

并非所有 Amazon EC2 Auto Scaling 操作均支持资源级权限。对于不支持资源级别权限的操作，您必须将通配符（*）作为资源。

以下 Amazon EC2 Auto Scaling 操作不支持资源级权限。

- DescribeAccountLimits
- DescribeAdjustmentTypes
- DescribeAutoScalingGroups
- DescribeAutoScalingInstances
- DescribeAutoScalingNotificationTypes
- DescribeInstanceRefreshes
- DescribeLaunchConfigurations
- DescribeLifecycleHooks
- DescribeLifecycleHookTypes
- DescribeLoadBalancers
- DescribeLoadBalancerTargetGroups
- DescribeMetricCollectionTypes
- DescribeNotificationConfigurations
- DescribePolicies
- DescribeScalingActivities
- DescribeScalingProcessTypes
- DescribeScheduledActions
- DescribeTags
- DescribeTerminationPolicyTypes
- DescribeWarmPool

Amazon EC2 Auto Scaling 的策略条件键

支持特定于服务的策略条件键	是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 Amazon 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则 Amazon 使用逻辑 OR 运算来评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM policy 元素：变量和标签。

Amazon支持全局条件键和特定于服务的条件键。要查看所有 Amazon 全局条件键，请参阅《IAM 用户指南》 中的 Amazon 全局条件上下文键。

Amazon EC2 Auto Scaling 定义了自己的一组条件键，还支持使用一些全局条件键。

Amazon EC2 Auto Scaling 支持以下条件键，您可以在权限策略中使用这些条件键来确定谁可以访问 Amazon EC2 Auto Scaling：

- autoscaling:InstanceTypes

- autoscaling:LaunchConfigurationName
- autoscaling:LaunchTemplateVersionSpecified
- autoscaling:LoadBalancerNames
- autoscaling:MaxSize
- autoscaling:MinSize
- autoscaling:ResourceTag/*key-name*: *tag-value*
- autoscaling:TargetGroupARNs
- autoscaling:VPCZoneIdentifiers

以下条件键特定于创建启动配置请求：

- autoscaling:ImageId
- autoscaling:InstanceType
- autoscaling:MetadataHttpEndpoint
- autoscaling:MetadataHttpPutResponseHopLimit
- autoscaling:MetadataHttpTokens
- autoscaling:SpotPrice

Amazon EC2 Auto Scaling 还支持以下全局条件键，您可以使用这些键根据请求中的标签或在自动扩缩组中提供的标签定义权限。有关更多信息，请参阅为 Auto Scaling 组和实例添加标签 (p. 87)。

- aws:RequestTag/*key-name*: *tag-value*
- aws:ResourceTag/*key-name*: *tag-value*
- aws:TagKeys: *[tag-key, ...]*

如需了解您可以与条件键搭配使用的 Amazon EC2 Auto Scaling API 操作，请参阅《服务授权参考》中的 Amazon EC2 Auto Scaling 定义的操作。有关使用 Amazon EC2 Auto Scaling 条件键的更多信息，请参阅 Amazon EC2 Auto Scaling 的条件键。

有关可用于控制访问权限的 IAM policy 示例，请参阅以下主题：

- 有关使用条件键控制对自动扩缩组和启动配置操作的访问权限的示例，请参阅 为 Auto Scaling 组和实例添加标签 (p. 87) 和 Amazon EC2 Auto Scaling 基于身份的策略示例 (p. 319)。
- 有关其他示例，包括若请求中已指定启动配置时拒绝访问自动扩缩组的示例，请参阅 启动模板支持 (p. 328)。

Amazon EC2 Auto Scaling 中的 ACL

支持 ACL	否

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

将 ABAC 与 Amazon EC2 Auto Scaling 结合使用

支持 ABAC（策略中的标签）	部分

基于属性的访问控制（ABAC）是一种授权策略，该策略基于属性来定义权限。在 Amazon 中，这些属性称为标签。您可以将标签附加到 IAM 实体（用户或角色）以及Amazon资源。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为 Yes（是）。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为 Partial（部分）。

有关 ABAC 的更多信息,请参阅《IAM 用户指南》中的什么是 ABAC？。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问控制（ABAC）。

ABAC 可用于支持标签的资源，但并非所有资源都支持标签。启动配置和扩缩策略不支持标签，但自动扩缩组支持标签。

有关为自动扩缩组添加标签的更多信息，请参阅为 Auto Scaling 组和实例添加标签 (p. 87)。

要查看基于身份的策略示例，了解如何基于自动扩缩组上的标签限制对该组的访问，请参阅安全性标签 (p. 91)。

将临时凭证与 Amazon EC2 Auto Scaling 一起使用

支持临时凭证	是

某些 Amazon Web Services在您使用临时凭证登录时无法正常工作。有关更多信息，包括 Amazon Web Services与临时凭证配合使用，请参阅 IAM 用户指南中的使用 IAM 的 Amazon Web Services。

如果您不使用用户名和密码而用其它方法登录到Amazon Web Services Management Console，则使用临时凭证。例如，当您使用贵公司的单点登录（SSO）链接访问Amazon时，该过程将自动创建临时凭证。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的切换到角色（控制台）。

您可以使用 Amazon CLI 或者 Amazon API 创建临时凭证。之后，您可以使用这些临时凭证访问 Amazon。Amazon建议您动态生成临时凭证，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

Amazon EC2 Auto Scaling 的服务角色

支持服务角色	是

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅 IAM 用户指南中的创建向 Amazon Web Service 委派权限的角色。

在创建会通知某个 Amazon SNS 主题或 Amazon SQS 队列的生命周期挂钩时，您必须指定一个角色，以允许 Amazon EC2 Auto Scaling 代表您访问 Amazon SNS 或 Amazon SQS。使用 IAM 控制台为您的生命周期挂钩设置服务角色。控制台可帮助您使用托管式策略创建一个具有足够权限集的角色。有关更多信息，请参阅 使用 Amazon SNS 接收通知 (p. 196) 和 使用 Amazon SQS 接收通知 (p. 197)：

创建自动扩缩组时，您可以选择传入一个服务角色，以允许 Amazon EC2 实例代表您访问其他 Amazon Web Services。Amazon EC2 实例的服务角色（对于启动模板或启动配置又称为 Amazon EC2 实例配置文

件）是一种特殊类型的服务角色，将在实例启动时分配给自动扩缩组中的每个 EC2 实例。您可以使用 IAM 控制台或 Amazon CLI 创建此服务角色。有关更多信息，请参阅在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)。

Warning

更改服务角色的权限可能会破坏 Amazon EC2 Auto Scaling 的功能。仅当 Amazon EC2 Auto Scaling 提供相关指导时才可编辑服务角色。

Amazon EC2 Auto Scaling 的服务相关角色

支持服务相关角色	是

服务相关角色是一种与 Amazon Web Service相关的服务角色。服务可以代入代表您执行操作的角色。服务相关角色显示在您的 Amazon Web Services 账户 中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理 Amazon EC2 Auto Scaling 服务相关角色的详细信息，请参阅 Amazon EC2 Auto Scaling 的服务相关角色 (p. 314)。

Amazon EC2 Auto Scaling 的 Amazon 托管策略

要向用户、组和角色添加权限，与自己编写策略相比，使用 Amazon 托管策略更简单。创建仅为团队提供所需权限的 IAM 客户托管策略需要时间和专业知识。要快速入门，您可以使用我们的 Amazon 托管式策略。这些策略涵盖常见使用案例，可在您的 Amazon Web Services 账户 中使用。有关 Amazon 托管式策略的更多信息，请参阅 IAM 用户指南中的Amazon 托管式策略。

Amazon Web Services 负责维护和更新 Amazon 托管式策略。您无法更改 Amazon 托管式策略中的权限。服务偶尔会向 Amazon 托管式策略添加额外权限以支持新功能。此类更新会影响附加策略的所有身份（用户、组和角色）。当启动新功能或新操作可用时，服务最有可能会更新 Amazon 托管式策略。服务不会从 Amazon 托管式策略中删除权限，因此策略更新不会破坏您的现有权限。

此外，Amazon 还支持跨多种服务的工作职能的托管式策略。例如，`ViewOnlyAccess` Amazon 托管式策略提供对许多 Amazon Web Services 服务和资源的只读访问权限。当服务启动新功能时，Amazon 会为新操作和资源添加只读权限。有关工作职能策略的列表和说明，请参阅 IAM 用户指南中的适用于工作职能的 Amazon 托管策略。

Amazon EC2 Auto Scaling 托管策略

可以将以下托管策略附加到 Amazon Identity and Access Management（IAM）身份（用户或角色）。每个策略提供 Amazon EC2 Auto Scaling 的全部或部分 API 操作的访问权限。

- AutoScalingFullAccess – 对需要从 Amazon CLI 或开发工具包完全访问 Amazon EC2 Auto Scaling 的 IAM 身份授予对 Amazon EC2 Auto Scaling 的完全访问权限，但不能访问 Amazon Web Services Management Console。
- AutoScalingReadOnlyAccess – 授予仅对 Amazon CLI 或开发工具包进行调用的 IAM 身份对 Amazon EC2 Auto Scaling 的只读访问权限。
- AutoScalingConsoleFullAccess - 使用 Amazon Web Services Management Console 授予对 Amazon EC2 Auto Scaling 的完全访问权限。此策略在您使用启动配置时有效，但在使用启动模板时不适用。
- AutoScalingConsoleReadOnlyAccess - 使用 Amazon Web Services Management Console 授予对 Amazon EC2 Auto Scaling 的只读访问权限。此策略在您使用启动配置时有效，但在使用启动模板时不适用。

当您使用控制台中的启动模板时，您需要授予特定于启动模板的其他权限，启动模板支持 (p. 328) 中对此进行了讨论。Amazon EC2 Auto Scaling 控制台需要 ec2 操作的以下权限，以便使用启动模板显示有关启动模板和启动实例的信息。

AutoScalingServiceRolePolicy Amazon 托管策略

无法将 AutoScalingServiceRolePolicy 附加到您的 IAM 身份。此策略附加到允许 Amazon EC2 Auto Scaling 启动和终止实例的服务相关角色。有关更多信息，请参阅Amazon EC2 Auto Scaling 的服务相关角色 (p. 314)。

Amazon EC2 Auto Scaling 更新为 Amazon 托管策略

查看有关 Amazon EC2 Auto Scaling 的 Amazon 托管策略的更新的详细信息（此服务开始跟踪这些更改）。有关此页面更改的自动提示，请订阅 Amazon EC2 Auto Scaling 文档历史记录页面上的 RSS 源。

更改	说明	日期
Amazon EC2 Auto Scaling 为其服务相关角色添加权限	AutoScalingServiceRolePolicy 策略现在授予服务访问与 VPC Lattice 集成所需的 API 操作的权限。 • GetTargetGroup 和 ListTargetGroup 操作。检索 VPC Lattice 目标组相关信息所需。 • RegisterTargets 和 DeregisterTargets 操作。在 VPC Lattice 目标组中注册和取消注册实例所需。 • ListTargets。允许 Amazon EC2 Auto Scaling 检索注册到 VPC Lattice 目标组的实例的运行状况信息。 有关更多信息，请参阅Amazon EC2 Auto Scaling 的服务相关角色 (p. 314)。	2022 年 12 月 6 日
Amazon EC2 Auto Scaling 为其服务相关角色添加权限	为了支持在创建启动模板时使用 Amazon Systems Manager 参数作为 AMI ID 的别名，AutoScalingServiceRolePolicy 策略现在授予调用 Amazon Systems Manager GetParameters API 操作的权限。有关更多信息，请参阅Amazon EC2 Auto Scaling 的服务相关角色 (p. 314)。	2022 年 3 月 28 日
Amazon EC2 Auto Scaling 为其服务相关角色添加权限	为支持预测式扩缩，AutoScalingServiceRolePolicy 策略现在包含调用 CloudWatch GetMetricData API 操作的权限。有关更多信息，请参阅Amazon	2021 年 5 月 19 日

更改	说明	日期
	EC2 Auto Scaling 的服务相关角色 (p. 314)。	
Amazon EC2 Auto Scaling 开始跟踪更改	Amazon EC2 Auto Scaling 已开始跟踪 Amazon 托管式策略的更改。	2021 年 5 月 19 日

Amazon EC2 Auto Scaling 的服务相关角色

Amazon EC2 Auto Scaling 使用服务相关角色来获取代表您调用其他 Amazon Web Services 所需的权限。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon Web Service 直接相关。

服务相关角色提供了一种将权限委托给其他 Amazon Web Services 的安全方式，因为只有相关服务才能担任服务相关角色。有关更多信息，请参阅 IAM 用户指南中的使用服务相关角色。服务相关角色还允许通过 Amazon CloudTrail 查看所有 API 调用。这样便于满足监控和审核要求，因为您可以跟踪 Amazon EC2 Auto Scaling 代表您执行的所有操作。有关更多信息，请参阅使用 Amazon CloudTrail 记录 Amazon EC2 Auto Scaling API 调用 (p. 256)。

下面的部分介绍如何创建和管理 Amazon EC2 Auto Scaling 服务相关角色。首先配置权限以允许 IAM 身份（如用户或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的使用服务相关角色。

目录

- 概览 (p. 314)
- 服务相关角色授予的权限 (p. 315)
- 创建服务相关角色（自动） (p. 317)
- 创建服务相关角色（手动） (p. 317)
- 编辑服务相关角色 (p. 318)
- 删除服务相关角色 (p. 318)
- Amazon EC2 Auto Scaling 服务相关角色支持的区域 (p. 319)

概览

有两种类型的 Amazon EC2 Auto Scaling 服务相关角色：

- 您账户的默认服务相关角色，名为 AWSServiceRoleForAutoScaling。此角色自动分配给 Auto Scaling 组，除非您指定不同的服务相关角色。
- 带有您在创建角色时指定的自定义后缀的服务相关角色，例如 AWSServiceRoleForAutoScaling_***mysuffix***。

自定义后缀服务相关角色的权限与默认的服务相关角色的权限完全相同。在这两种情况下，如果某个 Auto Scaling 组仍在使用这些角色，您不能编辑也不能删除它们。唯一的区别是角色名称后缀。

您可以在编辑 Amazon Key Management Service 密钥策略指定任一角色，以允许使用您的客户托管密钥对 Amazon EC2 Auto Scaling 启动的实例进行加密。但是，如果您计划对特定的客户托管密钥提供精细访问，应使用自定义后缀服务相关角色。使用自定义后缀服务相关角色可为您提供：

- 对客户托管密钥的更多控制
- 能够在 CloudTrail 日志中跟踪哪些 Auto Scaling 组进行了 API 调用

如果您创建并非所有用户都有权访问的客户托管密钥，请按照以下步骤操作，以允许使用自定义后缀服务相关角色：

1. 创建具有自定义后缀的服务相关角色。有关更多信息，请参阅创建服务相关角色（手动）(p. 317)。
2. 向服务相关角色授予对客户托管密钥的访问权限。有关允许服务相关角色使用密钥的密钥策略的更多信息，请参阅 与加密卷结合使用时必需的 Amazon KMS 密钥策略 (p. 333)。
3. 授予用户对您创建的服务相关角色的访问权限。有关创建 IAM policy 的更多信息，请参阅 控制能够传递哪些服务相关角色（使用 PassRole）(p. 325)。如果用户尝试指定服务相关角色而无权将该角色传递给服务，则会收到错误。

服务相关角色授予的权限

Amazon EC2 Auto Scaling 使用名为 AWSServiceRoleForAutoScaling 的服务相关角色或您的自定义后缀服务相关角色。

服务相关角色仅信任以下服务来担任该角色：

- autoscaling.amazonaws.com

此角色使用 AutoScalingServiceRolePolicy (p. 312) 策略，其中包含以下权限：

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"EC2InstanceManagement",
      "Effect":"Allow",
      "Action":[
        "ec2:AttachClassicLinkVpc",
        "ec2:CancelSpotInstanceRequests",
        "ec2:CreateFleet",
        "ec2:CreateTags",
        "ec2:DeleteTags",
        "ec2:Describe*",
        "ec2:DetachClassicLinkVpc",
        "ec2:ModifyInstanceAttribute",
        "ec2:RequestSpotInstances",
        "ec2:RunInstances",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource":"*"
    },
    {
      "Sid":"EC2InstanceProfileManagement",
      "Effect":"Allow",
      "Action":[
        "iam:PassRole"
      ],
      "Resource":"*",
      "Condition":{
        "StringLike":{
          "iam:PassedToService":"ec2.amazonaws.com*"
        }
      }
    },
    {
      "Sid":"EC2SpotManagement",
      "Effect":"Allow",
```

```
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"*",
      "Condition":{
        "StringEquals":{
          "iam:AWSServiceName":"spot.amazonaws.com"
        }
      }
    },
    {
      "Sid":"ELBManagement",
      "Effect":"Allow",
      "Action":[
        "elasticloadbalancing:Register*",
        "elasticloadbalancing:Deregister*",
        "elasticloadbalancing:Describe*"
      ],
      "Resource":"*"
    },
    {
      "Sid":"CWManagement",
      "Effect":"Allow",
      "Action":[
        "cloudwatch:DeleteAlarms",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricData",
        "cloudwatch:PutMetricAlarm"
      ],
      "Resource":"*"
    },
    {
      "Sid":"SNSManagement",
      "Effect":"Allow",
      "Action":[
        "sns:Publish"
      ],
      "Resource":"*"
    },
    {
      "Sid":"EventBridgeRuleManagement",
      "Effect":"Allow",
      "Action":[
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:DeleteRule",
        "events:DescribeRule"
      ],
      "Resource":"*",
      "Condition":{
        "StringEquals":{
          "events:ManagedBy":"autoscaling.amazonaws.com"
        }
      }
    },
    {
      "Sid":"SystemsManagerParameterManagement",
      "Effect":"Allow",
      "Action":[
        "ssm:GetParameters"
      ],
      "Resource": "*"
    },
    {
      "Sid":"VpcLatticeManagement",
```

```
      "Effect":"Allow",
      "Action":[
        "vpc-lattice:DeregisterTargets",
        "vpc-lattice:GetTargetGroup",
        "vpc-lattice:ListTargets",
        "vpc-lattice:ListTargetGroups",
        "vpc-lattice:RegisterTargets"
      ],
      "Resource": "*"
    }
  ]
}
```

此角色有权执行以下所有操作：

- `ec2` - 创建、描述、修改、启动/停止和终止 EC2 实例。
- `iam` - 将 IAM 角色传递给 (p. 331) EC2 实例，以便在实例上运行的应用程序可以访问该角色的临时凭证。
- `iam` - 创建 AWSServiceRoleForEC2Spot 服务相关角色以允许 Amazon EC2 Auto Scaling 代表您启动竞价型实例。
- `elasticloadbalancing` - 使用 Elastic Load Balancing 注册和取消注册实例，并检查注册目标的运行状况。
- `cloudwatch` - 创建、描述、修改和删除用于扩展策略的 CloudWatch 告警，并检索用于预测式扩展的指标。
- `sns` - 在实例启动或终止时向 Amazon SNS 发布通知。
- `events` - 代表您创建、描述、更新和删除 EventBridge 规则。
- `ssm` - 在启动模板中使用 Systems Manager 参数作为 AMI ID 的别名时，从 Parameter Store 中读取参数。
- `vpc-lattice` - 使用 VPC Lattice 注册和取消注册实例，并检查注册目标的运行状况。

创建服务相关角色（自动）

首次创建自动扩缩组时，Amazon EC2 Auto Scaling 会为您创建 AWSServiceRoleForAutoScaling 服务相关角色，除非您手动创建自定义后缀服务相关角色并在创建组时指定该角色。

> **Important**
>
> 您必须具备创建服务相关角色的 IAM 权限。否则，自动创建操作将失败。有关更多信息，请参阅 IAM 用户指南和本指南中的 创建服务相关角色所需的权限 (p. 325) 的服务相关角色权限。

2018 年 3 月，Amazon EC2 Auto Scaling 开始支持服务相关角色。如果您在此之前创建了自动扩缩组，则 Amazon EC2 Auto Scaling 会在您的账户中创建 AWSServiceRoleForAutoScaling 角色。有关更多信息，请参阅 IAM 用户指南中的我的 Amazon Web Services 账户 中出现新角色。

创建服务相关角色（手动）

创建服务相关角色 (控制台)

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择 Roles（角色）和 Create role（创建角色）。
3. 对于选择可信实体，选择 Amazon 服务。
4. 在选择将使用此角色的服务下，选择 EC2 Auto Scaling 和 EC2 Auto Scaling 使用案例。
5. 依次选择 Next: Permissions (下一步: 权限)、Next: Tags (下一步: 标签) 和 Next: Review (下一步: 审核)。注意：您无法在创建过程中将标签附加到服务相关角色。

6. 在 Review（审核）页面上，将 Role name（角色名称）留空，以便创建名称为 AWSServiceRoleForAutoScaling 的服务相关角色，或者输入后缀以创建名称为 AWSServiceRoleForAutoScaling_***suffix*** 的服务相关角色。
7. （可选）对于角色描述，编辑服务相关角色的描述。
8. 选择 Create role（创建角色）。

创建服务相关角色 (Amazon CLI)

使用以下 create-service-linked-role CLI 命令可创建名为 AWSServiceRoleForAutoScaling_***suffix*** 的适用于 Amazon EC2 Auto Scaling 的服务相关角色。

```
aws iam create-service-linked-role --aws-service-name autoscaling.amazonaws.com --custom-
suffix suffix
```

此命令的输出包含服务相关角色的 ARN，您可以用来向服务相关角色提供对客户托管密钥的访问权限。

```
{
    "Role": {
        "RoleId": "ABCDEF0123456789ABCDEF",
        "CreateDate": "2018-08-30T21:59:18Z",
        "RoleName": "AWSServiceRoleForAutoScaling_suffix",
        "Arn": "arn:aws:iam::123456789012:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling_suffix",
        "Path": "/aws-service-role/autoscaling.amazonaws.com/",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "sts:AssumeRole"
                    ],
                    "Principal": {
                        "Service": [
                            "autoscaling.amazonaws.com"
                        ]
                    },
                    "Effect": "Allow"
                }
            ]
        }
    }
}
```

有关更多信息，请参阅 IAM 用户指南中的创建服务相关角色。

编辑服务相关角色

您不能编辑为 Amazon EC2 Auto Scaling 创建的服务相关角色。创建服务相关角色后，您将无法更改角色的名称或其权限。但是，您可以编辑角色的说明。有关更多信息，请参阅 IAM 用户指南中的编辑服务相关角色。

Note

在 Amazon 中，不建议修改服务相关角色，因为这可能会导致跨服务混淆代理等安全问题。

删除服务相关角色

如果您不使用某个 Auto Scaling 组，我们建议您删除其服务相关角色。删除此角色会防止您拥有不使用或不主动监视和维护的实体。

只有在先删除相关资源后，才能删除服务相关角色。这可以防止您无意中撤销 Amazon EC2 Auto Scaling 对您的资源的权限。如果某个服务相关角色与多个 Auto Scaling 组结合使用，则必须删除使用该服务相关角色的所有 Auto Scaling 组，然后才能删除该服务相关角色。有关更多信息，请参阅删除 Auto Scaling 基础设施 (p. 117)。

您可以使用 IAM 删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的删除服务相关角色。

如果删除 AWSServiceRoleForAutoScaling 服务相关角色，则在您创建自动扩缩组但未指定其他服务相关角色时，Amazon EC2 Auto Scaling 将再次创建该角色。

Amazon EC2 Auto Scaling 服务相关角色支持的区域

Amazon EC2 Auto Scaling 支持在该服务可用的所有 Amazon Web Services 区域 中使用服务相关角色。

Amazon EC2 Auto Scaling 基于身份的策略示例

默认情况下，Amazon Web Services 账户 中的全新用户没有执行任何操作的权限。IAM 管理员必须创建并分配 IAM policy，以便为 IAM 身份（例如用户或角色）授予执行 Amazon EC2 Auto Scaling API 操作的权限。

要了解如何使用这些示例 JSON 策略文档创建 IAM policy，请参阅《IAM 用户指南》中的在 JSON 选项卡上创建策略。

下面介绍权限策略示例。

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:UpdateAutoScalingGroup",
          "autoscaling:DeleteAutoScalingGroup"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": { "autoscaling:ResourceTag/environment": "test" }
      }
   },
   {
      "Effect": "Allow",
      "Action": [
          "autoscaling:*LaunchConfiguration*",
          "autoscaling:Describe*"
      ],
      "Resource": "*"
   }]
}
```

此示例策略授予创建、修改和删除自动扩缩组的权限，但仅限于组使用标签 `environment=test` 时。由于启动配置不支持标签并且 `Describe` 操作不支持资源级权限，因此您必须在不带条件的单独语句中指定它们。要了解有关 IAM policy 语句中的元素的更多信息，请参阅 Amazon EC2 Auto Scaling 基于身份的策略 (p. 306)。

目录

- 策略最佳实践 (p. 320)
- 客户托管式策略示例 (p. 320)
    - 控制可以使用哪些标签键和标签值 (p. 321)

- 基于标签控制对 Auto Scaling 资源的访问 (p. 322)
  - 控制对创建和管理 Auto Scaling 组和扩展策略的访问 (p. 322)
  - 控制可以删除哪些扩展策略 (p. 322)
- 控制 Auto Scaling 组的最小容量和最大容量 (p. 323)
- 允许用户更改 Auto Scaling 组的容量 (p. 323)
- 允许用户创建和使用启动配置 (p. 324)

- 创建服务相关角色所需的权限 (p. 325)
  - 控制能够传递哪些服务相关角色（使用 PassRole）(p. 325)
- Amazon EC2 Auto Scaling 的其他权限 (p. 326)

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 Amazon EC2 Auto Scaling 资源。这些操作可能会使 Amazon Web Services 账户 产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- Amazon 托管策略及转向最低权限许可入门 - 要开始向用户和工作负载授予权限，请使用 Amazon 托管策略来为许多常见使用场景授予权限。您可以在 Amazon Web Services 账户 中找到这些策略。我们建议通过定义特定于您的使用场景的 Amazon 客户管理型策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 Amazon 托管策略或工作职能的 Amazon 托管策略。
- 应用最低权限 – 在使用 IAM policy 设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM policy 中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果通过特定 Amazon Web Service（例如 Amazon CloudFormation）使用服务操作，您还可以使用条件来授予对服务操作的访问权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM policy，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM policy语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM Acess Analyzer 策略验证。
- 需要多重身份验证（MFA）：如果您的 Amazon Web Services 账户 需要 IAM 用户或根用户，请启用 MFA 来提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实践的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

> Note
>
> 有些 Amazon EC2 Auto Scaling API 操作允许您在策略中包括该操作可以创建或修改的特定 Auto Scaling 组。您可以通过指定单个 Auto Scaling 组 ARN 来限制这些操作的目标资源。但是，作为最佳做法，我们建议您使用基于标签的策略，以允许（或拒绝）对具有特定标签的 Auto Scaling 组执行操作。

客户托管式策略示例

您可以创建自己的自定义 IAM policy，以允许或拒绝 IAM 身份（用户或角色）执行 Amazon EC2 Auto Scaling 操作的权限。您可以将这些自定义策略附加到需要指定权限的 IAM 身份。以下示例显示了几个常见使用情形的权限。

创建和更新 Auto Scaling 组时，某些操作要求执行某些其他操作。您还可以在 IAM policy 语句的 `Action` 元素中指定这些其他操作。例如，根据您想要提供的访问权限，可能还需要针对 Elastic Load Balancing、CloudWatch 和 Amazon SNS 的其他 API 操作。

主题

- 控制可以使用哪些标签键和标签值 (p. 321)
- 基于标签控制对 Auto Scaling 资源的访问 (p. 322)
- 控制 Auto Scaling 组的最小容量和最大容量 (p. 323)
- 允许用户更改 Auto Scaling 组的容量 (p. 323)
- 允许用户创建和使用启动配置 (p. 324)

控制可以使用哪些标签键和标签值

您可以在 IAM policy 中使用条件来控制可应用到 Auto Scaling 组的标签键和标签值。

要授予创建自动扩缩组或为该组添加标签（仅当请求者指定特定标签时）的权限，请使用 `aws:RequestTag` 条件键。要仅允许特定的标签键，请使用带 `aws:TagKeys` 修饰符的 `ForAllValues` 条件键。

以下策略需要请求者在请求中指定包含键 `environment` 的标签。`"?*"` 值强制对于标签键有某个值。要使用通配符，您必须使用 `StringLike` 条件运算符。

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:CreateOrUpdateTags"
      ],
      "Resource": "*",
      "Condition": {
          "StringLike": { "aws:RequestTag/environment": "?*" }
      }
   }]
}
```

以下策略指定请求者只能使用标签 `purpose=webserver` 和 `cost-center=cc123` 标记自动扩缩组，并且只允许 `purpose` 和 `cost-center` 标签（不能指定其他标签）。

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:CreateOrUpdateTags"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": {
              "aws:RequestTag/purpose": "webserver",
              "aws:RequestTag/cost-center": "cc123"
          },
          "ForAllValues:StringEquals": { "aws:TagKeys": ["purpose", "cost-center"] }
      }
   }]
}
```

以下策略需要请求者在请求中指定至少一个标签，并且仅允许 `cost-center` 和 `owner` 键。

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:CreateOrUpdateTags"
      ],
      "Resource": "*",
      "Condition": {
          "ForAnyValue:StringEquals": { "aws:TagKeys": ["cost-center", "owner"] }
      }
   }]
}
```

Note

对于条件，条件键不区分大小写，条件值区分大小写。因此，要强制标签键区分大小写，请使用 `aws:TagKeys` 条件键，其中标签键指定为条件中的值。

基于标签控制对 Auto Scaling 资源的访问

您还可以在 IAM policy 中提供标签信息，以便根据通过使用 `autoscaling:ResourceTag` 条件键附加到自动扩缩组的标签控制访问。

控制对创建和管理 Auto Scaling 组和扩展策略的访问

以下策略授权使用其名称中包含 `Scaling` 字符串的所有 Amazon EC2 Auto Scaling 操作，前提是自动扩缩组具有标签 `purpose=webserver`。由于 `Describe` 操作不支持资源级权限，因此，您必须在不带条件的单独语句中必须指定它们。

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": ["autoscaling:*Scaling*"],
      "Resource": "*",
      "Condition": {
          "StringEquals": { "autoscaling:ResourceTag/purpose": "webserver" }
      }
   },
   {
      "Effect": "Allow",
      "Action": "autoscaling:Describe*Scaling*",
      "Resource": "*"
   }]
}
```

控制可以删除哪些扩展策略

以下策略授权使用 `autoscaling:DeletePolicy` 操作删除扩展策略。但是，如果对其执行操作的 Auto Scaling 组具有 `environment=production` 标签，此策略也会拒绝操作。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
      "Effect": "Allow",
      "Action": "autoscaling:DeletePolicy",
```

```
      "Resource": "*"
   },
   {
      "Effect": "Deny",
      "Action": "autoscaling:DeletePolicy",
      "Resource": "*",
      "Condition": {
          "StringEquals": { "autoscaling:ResourceTag/environment": "production" }
      }
   }]
}
```

控制 Auto Scaling 组的最小容量和最大容量

Amazon EC2 Auto Scaling 允许您限制可以创建的 Auto Scaling 组的大小。以下策略授权创建和更新具有标签 `allowed=true` 的所有自动扩缩组的权限，只要请求者指定的最小大小不小于 1 或最大大小不大于 10。

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:UpdateAutoScalingGroup"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": { "autoscaling:ResourceTag/allowed": "true" },
          "NumericGreaterThanEqualsIfExists": { "autoscaling:MinSize": 1 },
          "NumericLessThanEqualsIfExists": { "autoscaling:MaxSize": 10 }
      }
   }]
}
```

允许用户更改 Auto Scaling 组的容量

以下策略授予用户使用 `SetDesiredCapacity` 和 `TerminateInstanceInAutoScalingGroup` API 操作的权限。`Resource` 元素使用通配符 (*) 来指示用户可以更改任何 Auto Scaling 组的容量。

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "autoscaling:SetDesiredCapacity",
          "autoscaling:TerminateInstanceInAutoScalingGroup"
      ],
      "Resource": "*"
   }]
}
```

如果您不使用标签控制对 Auto Scaling 组的访问，则可以调整前面的语句，以授予用户仅更改名称以 `devteam-` 开头的 Auto Scaling 组的容量的权限。有关指定 ARN 值的详细信息，请参阅 Amazon EC2 Auto Scaling 内基于资源的策略 (p. 307)。

```
      "Resource":
 "arn:aws:autoscaling:region:123456789012:autoScalingGroup:*:autoScalingGroupName/devteam-
*"
```

您还可以通过将多个 ARN 包含在列表中来指定它们。包括 UUID 可确保将访问权授予特定的 Auto Scaling 组。新组的 UUID 与删除的同名组的 UUID 不同。

```
      "Resource": [

 "arn:aws:autoscaling:region:123456789012:autoScalingGroup:7fe02b8e-7442-4c9e-8c8e-85fa99e9b5d9:autoSca
devteam-1",
          "arn:aws:autoscaling:region:123456789012:autoScalingGroup:9d8e8ea4-22e1-44c7-
a14d-520f8518c2b9:autoScalingGroupName/devteam-2",
          "arn:aws:autoscaling:region:123456789012:autoScalingGroup:60d6b363-
ae8b-467c-947f-f1d308935521:autoScalingGroupName/devteam-3"
      ]
```

允许用户创建和使用启动配置

以下策略授权用户在实例类型为 t2.micro 且仅当启动配置的名称以 qateam- 开头时才创建启动配置。有关指定 ARN 值的详细信息，请参阅 Amazon EC2 Auto Scaling 内基于资源的策略 (p. 307)。这些策略可以仅为名称以 qateam- 开头的 Auto Scaling 组指定启动配置。

此语句的最后一部分授予用户描述启动配置和访问其账户中的某些 Amazon EC2 资源的权限。这为用户提供了从 Amazon EC2 Auto Scaling 控制台创建和管理启动配置的最低权限。

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": "autoscaling:CreateLaunchConfiguration",
      "Resource":
 "arn:aws:autoscaling:region:123456789012:launchConfiguration:*:launchConfigurationName/
qateam-*",
      "Condition": {
          "StringEquals": { "autoscaling:InstanceType": "t2.micro" }
      }
   },
   {
      "Effect": "Allow",
      "Action": [
          "autoscaling:CreateAutoScalingGroup",
          "autoscaling:UpdateAutoScalingGroup"
      ],
      "Resource": "*",
      "Condition": {
          "StringLikeIfExists": { "autoscaling:LaunchConfigurationName": "qateam-*" }
      }
   },
   {
      "Effect": "Allow",
      "Action": [
          "autoscaling:DescribeLaunchConfigurations",
          "ec2:DescribeImages",
          "ec2:DescribeVolumes",
          "ec2:DescribeInstances",
          "ec2:DescribeInstanceAttribute",
          "ec2:DescribeKeyPairs",
          "ec2:DescribeSecurityGroups",
          "ec2:DescribeSpotInstanceRequests",
          "ec2:DescribeSpotPriceHistory",
          "ec2:DescribeVpcClassicLink",
          "ec2:DescribeVpcs",
          "ec2:DescribeSubnets"
       ],
       "Resource": "*"
```

```
   }]
}
```

您可以向此策略添加 API 操作以便为用户提供更多选项，例如：

- `iam:ListInstanceProfiles`：列出实例配置文件。
- `ec2:CreateSecurityGroup`: 创建新安全组。
- `ec2:AuthorizeSecurityGroupIngress`：添加入站规则。
- `ec2:CreateKeyPair`：创建新的密钥对。

创建服务相关角色所需的权限

当您的 Amazon Web Services 账户 中的任何用户首次调用 Amazon EC2 Auto Scaling API 操作时，Amazon EC2 Auto Scaling 需要具有创建服务相关角色的权限。如果服务相关角色尚不存在，Amazon EC2 Auto Scaling 会在您的账户中创建此角色。此服务相关角色向 Amazon EC2 Auto Scaling 授予权限，以便其代表您调用其他 Amazon Web Services。

为使自动角色创建操作成功，用户必须具有 `iam:CreateServiceLinkedRole` 操作的权限。

```
"Action": "iam:CreateServiceLinkedRole"
```

下面显示了允许用户为 Amazon EC2 Auto Scaling 创建 Amazon EC2 Auto Scaling 服务相关角色的权限策略示例。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName":"autoscaling.amazonaws.com"
                }
            }
        }
    ]
}
```

控制能够传递哪些服务相关角色（使用 PassRole）

创建或更新自动扩缩组并在请求中指定自定义后缀服务相关角色的用户需要 `iam:PassRole` 权限。

如果您为不同的服务相关角色提供对不同密钥的访问权限，则可以使用 `iam:PassRole` 权限来保护 Amazon KMS 客户托管密钥的安全性。根据您组织的需求，您可能有三个密钥分别供开发团队、QA 团队和财务团队使用。首先，创建对所需密钥具有访问权限的服务相关角色，例如，名为 AWSServiceRoleForAutoScaling_devteamkeyaccess 的服务相关角色。然后，将策略附加到 IAM 身份，例如用户或角色。

以下策略授予权限，将 AWSServiceRoleForAutoScaling_devteamkeyaccess 角色传递给名称以 `devteam-` 开头的任意自动扩缩组。如果创建自动扩缩组的 IAM 身份尝试指定另一个与服务相关的角色，则会收到错误。如果用户选择不指定服务相关角色，则改为使用默认 AWSServiceRoleForAutoScaling 角色。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling_devteamkeyaccess",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "autoscaling.amazonaws.com"
                    ]
                },
                "StringLike": {
                    "iam:AssociatedResourceARN": [

 "arn:aws:autoscaling:region:123456789012:autoScalingGroup:*:autoScalingGroupName/devteam-
*"
                    ]
                }
            }
        }
    ]
}
```

有关自定义后缀服务相关角色的更多信息，请参阅 Amazon EC2 Auto Scaling 的服务相关角色 (p. 314)。

Amazon EC2 Auto Scaling 的其他权限

除 Amazon EC2 Auto Scaling 权限外，用户必须具有来自 Amazon EC2 和 IAM 的以下权限才能成功执行指定操作。

创建自动扩缩组（**autoscaling:CreateAutoScalingGroup**）

- iam:CreateServiceLinkedRole（如果您使用的是默认服务相关角色并且该角色尚不存在，则必需）
- iam:PassRole（如果您正在使用非默认服务相关角色，您正在为生命周期钩子的 RoleARN 参数指定 IAM 角色，或者您正在使用启动模板，而该启动模板指定可供 Auto Scaling 组中的实例使用的 IAM 角色，则需要此选项）
- ec2:RunInstance（如果您正在使用启动模板，则需要此选项）
- ec2:CreateTags（如果您正在使用启动模板，而该启动模板指定要应用于您的实例和 EBS 卷的标签，则需要此选项）

创建启动配置（**autoscaling:CreateLaunchConfiguration**）

- ec2:DescribeImages
- ec2:DescribeInstances
- ec2:DescribeInstanceAttribute
- ec2:DescribeKeyPairs
- ec2:DescribeSecurityGroups
- ec2:DescribeSpotInstanceRequests
- ec2:DescribeVpcClassicLink
- iam:PassRole（当您要为 IamInstanceProfile 参数指定 IAM 角色时需要)

创建生命周期挂钩（**autoscaling:PutLifecycleHook**）

- iam:PassRole（仅当您要为 RoleARN 参数指定 IAM 角色时需要)

防止跨服务混淆代理

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。

在 Amazon 中，跨服务模拟可能会导致混淆代理问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。

为了防止这种情况，Amazon 提供可帮助您保护所有服务的服务委托人数据的工具，这些服务委托人有权限访问账户中的资源。我们建议使用 Amazon EC2 Auto Scaling 服务角色信任策略中的 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文键。这些密钥限制了 Amazon EC2 Auto Scaling 为资源提供另一项服务的权限。

要使用 `aws:SourceArn` 或 `aws:SourceAccount` 全局条件键，将值设置为 Amazon 资源名称（ARN）或 Amazon EC2 Auto Scaling 存储的资源账户。请尽可能使用更具体的 `aws:SourceArn`。将值设置为 ARN 或带通配符 (*) 的 ARN 模式，用于 ARN 的未知部分。如果您不知道资源的 ARN，请改用 `aws:SourceAccount`。

以下示例演示如何使用 Amazon EC2 Auto Scaling 中的 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文键来防范混淆代理问题。

示例：使用 aws:SourceArn 和 aws:SourceAccount 条件键

由一项服务担任、代表您执行操作的角色称为服务角色 (p. 311)。如果您希望创建会将通知发送到除 Amazon EventBridge 以外任何地方的生命周期挂钩，您必须创建一个服务角色，以允许 Amazon EC2 Auto Scaling 代表您向 Amazon SNS 主题或 Amazon SQS 队列发送通知。如果您只希望将一个自动扩缩组与跨服务访问相关联，则可以指定服务角色的信任策略，如下所示。

此示例信任策略使用条件语句，将服务角色的 `AssumeRole` 功能限制为只能执行影响指定账户中指定自动扩缩组的操作。`aws:SourceArn` 和 `aws:SourceAccount` 条件会得到独立评估。使用服务角色的任何请求都必须满足这两个条件。

在使用此策略之前，请将区域、账户 ID、UUID 和组名称替换为您账户中的有效值。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "autoscaling.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn":
 "arn:aws:autoscaling:region:account_id:autoScalingGroup:uuid:autoScalingGroupName/my-asg"
      },
      "StringEquals": {
        "aws:SourceAccount": "account_id"
      }
    }
  }
}
```

其他信息

有关更多信息，请参阅《IAM 用户指南》中的 Amazon 全局条件上下文键、混淆代理人问题以及修改角色信任策略（控制台）。

启动模板支持

Amazon EC2 Auto Scaling 支持将 Amazon EC2 启动模板与您的 Auto Scaling 组一起使用。我们建议您允许用户从启动模板创建 Auto Scaling 组，因为这样做允许用户使用 Amazon EC2 Auto Scaling 和 Amazon EC2 的最新功能。例如，用户必须指定启动模板才能使用混合实例策略。

您可以使用 AmazonEC2FullAccess 策略授予用户使用其账户中的 Amazon EC2 Auto Scaling 资源、启动模板和其他 EC2 资源的完全权限。或者，您可以创建自己的自定义 IAM policy，为用户授予使用启动模板的精细访问权限，如本主题所述。

您可以为自己使用量身定制的示例策略

下面显示您可以为自身使用量身定制的基础权限策略示例。此策略授予创建、修改和删除所有自动扩缩组的权限，但仅限于组使用标签 environment=test 时。然后，它授予所有 Describe 操作的权限。由于 Describe 操作不支持资源级权限，因此，您必须在不带条件的单独语句中必须指定它们。

具有此策略的 IAM 身份（用户或角色）有权使用启动模板创建或更新自动扩缩组，因为他们还有权使用 ec2:RunInstances 操作。

```
{
    "Version": "2012-10-17",
    "Statement": [
          {
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:DeleteAutoScalingGroup"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": { "autoscaling:ResourceTag/environment": "test" }
            }
         },
         {
            "Effect": "Allow",
            "Action": [
                "autoscaling:Describe*",
                "ec2:RunInstances"
            ],
            "Resource": "*"
        }
    ]
}
```

当使用启动模板创建或更新 Auto Scaling 组时，将选中 ec2:RunInstances。使用此示例策略来构建您自己的限制 ec2:RunInstances 权限的策略，例如，通过要求使用带有特定标签的启动模板。

以下示例显示了您可用于控制当用户使用启动模板时具有的权限的策略语句。

目录

- 需要具有特定标签的启动模板 (p. 329)
- 需要启动模板和版本号 (p. 329)
- 要求使用实例元数据服务版本 2 (IMDSv2) (p. 329)
- 限制对 Amazon EC2 资源的访问 (p. 330)
- 标记实例和卷所需的权限 (p. 331)
- 其他权限 (p. 331)

- 了解更多信息 (p. 331)

需要具有特定标签的启动模板

以下示例限制对使用位于指定区域中且具有标签 environment=test 的启动模板调用 ec2:RunInstances 操作的访问权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:region:123456789012:launch-template/*",
            "Condition": {
                "StringEquals": { "ec2:ResourceTag/environment": "test" }
            }
        }
    ]
}
```

需要启动模板和版本号

如果他们指定了启动模板的版本号，然后拒绝使用启动配置创建或修改任何 Auto Scaling 组的权限，以下示例允许用户创建和修改 Auto Scaling 组。如果具有此策略的用户忽略版本号以指定 Latest 或 Default 启动模板版本，或者改为指定启动配置，操作将失败。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": { "autoscaling:LaunchTemplateVersionSpecified": "true" }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup"
            ],
            "Resource": "*",
            "Condition": {
                "Null": { "autoscaling:LaunchConfigurationName": "false" }
            }
        }
    ]
}
```

要求使用实例元数据服务版本 2 (IMDSv2)

为了提高安全性，您可以将用户的权限设置为要求使用需要 IMDSV2 的启动模板。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的配置实例元数据服务。

以下示例指定用户不能调用 ec2:RunInstances 操作，除非该实例也选择需要使用 IMDSv2（由 "ec2:MetadataHttpTokens":"required" 指示）。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireImdsV2",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringNotEquals": { "ec2:MetadataHttpTokens": "required" }
            }
        }
    ]
}
```

Tip

要强制替换 Auto Scaling 使用新启动模板或配置了实例元数据选项的启动模板的新版本启动模板启动，您可以终止组中的现有实例。Amazon EC2 Auto Scaling 会立即开始启动新实例以替换您终止的实例。或者，您可以启动实例刷新以执行组的滚动更新。有关更多信息，请参阅基于实例刷新替换 Auto Scaling 实例 (p. 95)。

限制对 Amazon EC2 资源的访问

以下示例通过限制对 Amazon EC2 资源的访问来控制用户可以启动的实例的配置。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:123456789012:subnet/subnet-1a2b3c4d",
                "arn:aws:ec2:region:123456789012:security-group/sg-903004f88example",
                "arn:aws:ec2:region:123456789012:network-interface/*",
                "arn:aws:ec2:region:123456789012:volume/*",
                "arn:aws:ec2:region::image/ami-04d5cc9b88example"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:region:123456789012:instance/*",
            "Condition": {
                "StringEquals": { "ec2:InstanceType": "t2.micro" }
            }
        }
    ]
}
```

在此示例中，有两个语句：

- 第一条语句要求用户使用特定安全组 (subnet-1a2b3c4d) 并使用特定 AMI (sg-903004f88example) 将实例启动到特定子网 (ami-04d5cc9b88example) 中。它还允许用户访问启动实例所需的额外资源：网络接口和卷。
- 第二个语句允许用户仅启动特定实例类型 (t2.micro) 的实例。

标记实例和卷所需的权限

以下示例允许用户在创建时标记实例和卷。如果在启动模板中指定了标签，则需要此策略。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的在创建过程中授予标记资源的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:region:123456789012:*/*",
            "Condition": {
                "StringEquals": { "ec2:CreateAction": "RunInstances" }
            }
        }
    ]
}
```

其他权限

根据要支持的方案，您可以在 IAM policy 语句的 `Action` 元素中指定这些附加操作。

您必须向控制台用户授予对 `ec2:DescribeLaunchTemplates` 和 `ec2:DescribeLaunchTemplateVersions` 操作的权限。如果没有这些权限，则无法在 Auto Scaling 组向导中加载启动模板数据，并且用户无法逐步通过向导使用启动模板启动实例。

在使用启动模板时，您还必须授予创建、修改、描述和删除启动模板和启动模板版本的权限。您可以通过控制对 `ec2:CreateLaunchTemplate` 和 `ec2:CreateLaunchTemplateVersion` 操作的访问来控制谁可以创建启动模板和启动模板版本。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的控制启动模板的使用和示例：使用启动模板。

> **Note**
>
> 对于配置为使用 `Latest` 或 `Default` 启动模板版本的自动扩缩组，在创建启动模板的新版本时，Amazon EC2 Auto Scaling 不会检查在启动实例时要完成的操作的权限。在为可以创建和管理启动模板版本的用户设置权限时，这是一个重要的考虑因素。

除了 Amazon EC2 操作外，使用指定实例配置文件（IAM 角色的容器）的启动模板创建或更新自动扩缩组的用户还需要 `iam:PassRole` 权限。有关更多信息和示例 IAM policy，请参阅 在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)。

以使用 Amazon Systems Manager 参数的启动模板创建或更新自动扩缩组的用户需要 `ssm:GetParameters` 权限。有关更多信息，请参阅在启动模板中使用 Amazon Systems Manager 参数代替 AMI ID (p. 35)。

了解更多信息

在使用 IAM 管理对启动模板的访问之前，您首先应了解哪些 IAM 功能可与启动模板结合使用。有关更多信息，请参阅《服务授权参考》中的 Amazon EC2 的操作、资源和条件键。

在 Amazon EC2 实例上运行的应用程序的 IAM 角色

在 Amazon EC2 实例上运行的应用程序需要凭证以访问其他 Amazon Web Services。要以安全的方式提供这些凭证，请使用 IAM 角色。角色可提供临时权限，以供应用程序在访问其他 Amazon 资源时使用。角色的权限将确定允许访问资源的应用程序。

对于 Auto Scaling 组中的实例，您必须创建启动模板或启动配置，并选择要与实例关联的实例配置文件。实例配置文件是 IAM 角色的容器，在实例启动时允许 Amazon EC2 将 IAM 角色传递到实例。首先，创建具有访问 Amazon 资源所需的所有权限的 IAM 角色。然后，创建实例配置文件并将该角色分配给它。

> **Note**
>
> 作为一项最佳实践，我们强烈建议您创建角色，使其具有您的应用程序需要的其他 Amazon Web Services 的最小权限。

目录

- 先决条件 (p. 332)
- 创建启动模板 (p. 333)
- 创建启动配置 (p. 333)
- 另请参阅 (p. 333)

先决条件

创建运行在 Amazon EC2 上的应用程序可代入的 IAM 角色。选择适当的权限，以便随后向应用程序提供可以进行所需 API 调用的角色。

如果您使用 IAM 控制台而不是 Amazon CLI 或其中一个 Amazon SDK，控制台自动创建实例配置文件，按相应的角色为文件命名。

创建 IAM 角色（控制台）

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在左侧的导航窗格中，选择角色。
3. 选择 Create role（创建角色）。
4. 对于选择可信实体，选择 Amazon 服务。
5. 对于您的使用案例，选择 EC2，然后选择 Next（下一步）。
6. 如果可能，选择要用作权限策略的策略，或选择 Create policy（创建策略）以打开新的浏览器选项卡并从头开始创建新策略。有关更多信息，请参阅 IAM 用户指南 中的创建 IAM policy。在您创建策略后，关闭该选项卡并返回到您的原始选项卡。选中您希望服务具有的权限策略旁边的复选框。
7. （可选）设置权限边界。这是一项可用于服务角色的高级功能。有关更多信息，请参阅《IAM 用户指南》中的 IAM 实体的权限边界。
8. 选择 Next（下一步）。
9. 在 Name, review, and create（命名、检查并创建）页面上，对于 Role name（角色名称），请输入一个角色名称以帮助标识此角色的作用。此名称在您的 Amazon Web Services 账户中必须唯一。由于其他 Amazon 资源可能会引用此角色，因此角色创建完毕后将无法编辑角色名称。
10. 检查该角色，然后选择创建角色。

IAM 权限

使用基于 IAM 身份的策略来控制对新 IAM 角色的访问。使用指定实例配置文件的启动模板创建或更新自动扩缩组，或创建指定实例配置文件的启动配置的 IAM 身份（用户或角色）将需要 `iam:PassRole` 权限。

下面的策略示例授予仅传递名称以 `qateam-` 开头的 IAM 角色的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/qateam-*",
```

```
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn"
                    ]
                }
            }
        }
    ]
}
```

创建启动模板

当您使用 Amazon Web Services Management Console 创建启动模板时，在高级详细信息部分中，从 IAM 实例配置文件中选择角色。有关更多信息，请参阅配置启动模板的高级设置 (p. 24)。

当您使用 create-launch-template 命令从 Amazon CLI 中创建启动模板时，指定 IAM 角色的实例配置文件名称，如以下示例所示。

```
aws ec2 create-launch-template --launch-template-name my-lt-with-instance-profile --
version-description version1 \
--launch-template-data
 '{"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro","IamInstanceProfile":
{"Name":"my-instance-profile"}}'
```

创建启动配置

使用 Amazon Web Services Management Console 创建启动配置时，在其他配置部分中，从 IAM 实例配置文件中选择该角色。有关更多信息，请参阅创建启动配置 (p. 39)。

当您使用 create-launch-configuration 命令从 Amazon CLI 中创建启动配置时，指定 IAM 角色的实例配置文件名称，如以下示例所示。

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-with-
instance-profile \
--image-id ami-04d5cc9b88example --instance-type t2.micro \
--iam-instance-profile my-instance-profile
```

另请参阅

有关帮助您开始了解和使用 Amazon EC2 的 IAM 角色的更多信息，请参阅：

- 适用于 Linux 实例的 Amazon EC2 用户指南中的 Amazon EC2 的 IAM 角色
- IAM 用户指南中的使用使用实例配置文件和使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限

与加密卷结合使用时必需的 Amazon KMS 密钥策略

Amazon EC2 Auto Scaling 使用服务相关角色 (p. 314)将权限委托给其他 Amazon Web Services。Amazon EC2 Auto Scaling 服务相关角色经过预定义，并包含 Amazon EC2 Auto Scaling 代表您调用其他 Amazon Web Services 所需的权限。预定义的权限还包括对 Amazon 托管式密钥 的访问权限。但是，它们不包括对客户管理密钥的访问权限，因此您可以保持对这些密钥的完全控制。

本主题介绍当您为 Amazon EBS 加密指定客户托管密钥时如何设置启动 Auto Scaling 实例所需的密钥策略。

Note

Amazon EC2 Auto Scaling 无需额外授权即可使用默认 Amazon 托管式密钥 来保护您账户中的加密卷。

目录

- 概览 (p. 334)
- 配置密钥策略 (p. 334)
- 示例 1：允许访问客户托管密钥的关键策略部分 (p. 335)
- 示例 2：允许跨账户访问客户托管密钥的关键策略部分 (p. 335)
- 在 Amazon KMS 控制台中编辑密钥策略 (p. 337)

概览

当 Amazon EC2 Auto Scaling 启动实例时，以下 Amazon KMS keys 可用于 Amazon EBS 加密：

- Amazon 托管式密钥 — 您的账户中 Amazon EBS 创建、拥有和管理的加密密钥。这是新账户的默认加密密钥。Amazon 托管式密钥 用于加密，除非您指定客户托管密钥。
- 客户托管密钥 — 您创建、拥有和管理的自定义加密密钥。有关更多信息，请参阅 Amazon Key Management Service 开发人员指南中的创建密钥。

  注意：密钥必须是对称的。Amazon EBS 不支持非对称客户托管密钥。

在创建加密的快照或指定加密卷的启动模板，或者默认启用加密时，您可以配置客户托管密钥。

配置密钥策略

您的 KMS 密钥必须具有允许 Amazon EC2 Auto Scaling 启动使用 Amazon EBS 卷使用客户托管密钥加密的实例的密钥策略。

使用此页面上的示例配置密钥策略来向 Amazon EC2 Auto Scaling 提供访问客户托管密钥的权限。您可以在创建密钥时或以后的某个时间修改客户托管密钥的密钥策略。

您必须向您的密钥策略中添加至少两个策略语句，以便此策略可用于 Amazon EC2 Auto Scaling。

- 第一条语句允许在 `Principal` 元素中指定的 IAM 身份直接使用客户托管密钥。它包括对密钥执行 Amazon KMS `Encrypt`、`Decrypt`、`ReEncrypt*`、`GenerateDataKey*` 和 `DescribeKey` 操作的权限。
- 第二个语句允许在 `Principal` 元素中指定的 IAM 身份使用授权，以将其自身权限的某个子集委托给与 Amazon Web Services 集成的 Amazon KMS 服务或其他主体。这样，他们可以使用密钥代表您创建加密的资源。

当您向密钥策略添加新的策略语句时，不要更改策略中任何已存在的语句。

对于以下每个示例，必须替换的参数（如密钥 ID 或服务相关角色的名称）显示为`#######`。在大多数情况下，您可以使用 Amazon EC2 Auto Scaling 服务相关角色的名称替换此服务相关角色的名称。但是，当使用启动配置启动竞价型实例时，请使用名为 AWSServiceRoleForEC2Spot 的角色。

有关更多信息，请参阅以下资源：

- 要使用 Amazon CLI 创建密钥，请参阅 create-key。
- 要使用 Amazon CLI 更新密钥策略，请参阅 put-key-policy。
- 要查找密钥 ID 和 Amazon Resource Name (ARN)，请参阅 Amazon Key Management Service 开发人员指南中的查找密钥 ID 和 ARN。

- 有关 Amazon EC2 Auto Scaling 服务相关角色的信息，请参阅 Amazon EC2 Auto Scaling 的服务相关角色 (p. 314)。
- 有关 Amazon EBS 和 KMS 的更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南和 Amazon Key Management Service 开发人员指南中的 Amazon EBS 加密。

示例 1：允许访问客户托管密钥的关键策略部分

将以下两个策略语句添加到客户托管密钥的密钥策略中，同时将示例 ARN 替换为允许访问密钥的相应服务相关角色的 ARN。在本示例中，策略部分授予名为 AWSServiceRoleForAutoScaling 的服务相关角色使用客户托管密钥的权限。

```
{
   "Sid": "Allow service-linked role use of the customer managed key",
   "Effect": "Allow",
   "Principal": {
       "AWS": [
           "arn:aws:iam::123456789012:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
       ]
   },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
   ],
   "Resource": "*"
}
```

```
{
   "Sid": "Allow attachment of persistent resources",
   "Effect": "Allow",
   "Principal": {
       "AWS": [
           "arn:aws:iam::123456789012:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
       ]
   },
   "Action": [
       "kms:CreateGrant"
   ],
   "Resource": "*",
   "Condition": {
       "Bool": {
           "kms:GrantIsForAWSResource": true
       }
    }
}
```

示例 2：允许跨账户访问客户托管密钥的关键策略部分

如果您在与自动扩缩组不同的账户中创建了客户管理型密钥，则必须组合使用授权和密钥策略以允许对该密钥的跨账户存取。

必须按以下顺序完成两个步骤：

1. 首先，将以下两个策略语句添加到客户管理型密钥的密钥策略中。将示例 ARN 替换为另一个账户的 ARN，务必要将 *111122223333* 替换为您要在其中创建自动扩缩组的 Amazon Web Services 账户的实

际账户 ID。这将允许您向指定账户中的 IAM 用户或角色授予使用下面的 CLI 命令为密钥创建授权的权限。但这本身并不会向任何用户授予对密钥的访问权限。

```
{
   "Sid": "Allow external account 111122223333 use of the customer managed key",
   "Effect": "Allow",
   "Principal": {
       "AWS": [
           "arn:aws:iam::111122223333:root"
       ]
   },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
   ],
   "Resource": "*"
}
```

```
{
   "Sid": "Allow attachment of persistent resources in external account 111122223333",
   "Effect": "Allow",
   "Principal": {
       "AWS": [
           "arn:aws:iam::111122223333:root"
       ]
   },
   "Action": [
       "kms:CreateGrant"
   ],
   "Resource": "*"
}
```

2. 然后从您要在其中创建自动扩缩组的账户中，创建一个授权以将相关权限委托给相应的服务相关角色。该授权的 `Grantee Principal` 元素是相应的服务相关角色的 ARN。`key-id` 是密钥的 ARN。

   以下是一个示例 create-grant CLI 命令，它向账户 *111122223333* 中名为 AWSServiceRoleForAutoScaling 的服务相关角色授予在账户 *444455556666* 中使用客户托管密钥的权限。

```
aws kms create-grant \
  --region us-west-2 \
  --key-id arn:aws:kms:us-west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d \
  --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling \
  --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"
```

   要使此命令成功，发出请求的用户必须具有执行 `CreateGrant` 操作的权限。

   以下 IAM policy 示例允许账户 *111122223333* 中的 IAM 身份（用户或角色）为账户 *444455556666* 中的客户托管密钥创建授权。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreationOfGrantForTheKMSKeyinExternalAccount444455556666",
      "Effect": "Allow",
```

```
      "Action": "kms:CreateGrant",
      "Resource": "arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    }
  ]
}
```

有关为不同 Amazon Web Services 账户中的 KMS 密钥创建授权的更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的 Amazon KMS 中的授权。

Important

指定为被授权者委托人的服务相关角色名称必须是现有角色的名称。创建授权后，为确保该授权允许 Amazon EC2 Auto Scaling 使用指定的 KMS 密钥，请勿删除并重新创建服务相关角色。

在 Amazon KMS 控制台中编辑密钥策略

之前部分中的示例仅显示如何向密钥策略添加语句，这只是更改密钥策略的一种方法。更改密钥政策的最简单方法是使用 Amazon KMS 控制台的默认密钥政策视图，并使 IAM 身份（用户或角色）成为适当密钥政策的关键用户之一。有关更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的使用 Amazon Web Services Management Console 默认视图。

Important

请务必谨慎。控制台的默认视图策略语句包括对客户托管密钥执行 Amazon KMS Revoke 操作的权限。如果您授予 Amazon Web Services 账户 访问您账户中的客户托管密钥的权限，而您意外撤销了授予其此权限的授权，则外部用户将无法再访问其加密的数据或用于加密其数据的密钥。

Amazon EC2 Auto Scaling 的合规性验证

作为多个 Amazon 合规性计划的一部分，第三方审计员将评估 Amazon EC2 Auto Scaling 的安全性和合规性。其中包括 SOC、PCI、FedRAMP、HIPAA 及其它。

要了解某个 Amazon Web Service 是否在特定合规性计划范围内，请参阅 合规性计划范围内的 Amazon Web Services，然后选择您感兴趣的合规性计划。有关常规信息，请参阅Amazon Web Services合规性计划。

您可以使用 Amazon Artifact 下载第三方审计报告。有关更多信息，请参阅、在 Amazon Artifact 中下载报告。

您使用 Amazon Web Services 的合规性责任取决于您数据的敏感度、贵公司的合规性目标以及适用的法律法规。Amazon 提供以下资源来帮助满足合规性：

- 安全性与合规性快速入门指南 – 这些部署指南讨论了架构注意事项，并提供了在 Amazon 上部署以安全性和合规性为重点的基准环境的步骤。
- Amazon合规性资源 – 此业务手册和指南集合可能适用于您的行业和位置。
- Amazon Config 开发人员指南中的使用规则评估资源 – 此 Amazon Config 服务评估您的资源配置对内部实践、行业指南和法规的遵循情况。
- Amazon Security Hub：此 Amazon Web Service 提供了 Amazon 中安全状态的全面视图，可帮助您检查是否符合安全行业标准和最佳实践规范。

PCI DSS 合规性

Amazon EC2 Auto Scaling 支持由商家或服务提供商处理、存储和传输信用卡数据，而且已经验证符合支付卡行业 (PCI) 数据安全标准 (DSS)。有关 PCI DSS 的更多信息，包括如何请求 Amazon PCI Compliance Package 的副本，请参阅 PCI DSS 第 1 级。

Amazon EC2 Auto Scaling 中的恢复功能

Amazon 全球基础设施围绕 Amazon Web Services 区域 和可用区构建。Amazon Web Services 区域 提供多个在物理上独立且隔离的可用区，这些可用区与延迟率低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 Amazon Web Services 区域 和可用区的更多信息，请参阅 Amazon 全球基础设施。

要从可用区设计的地理冗余中获益，请执行以下操作：

- 在多个可用区之间跨越您的自动扩缩组。
- 在每个可用区内至少维护一个实例。
- 附加负载均衡器以在同一可用区之间分配传入流量。如果您使用应用程序负载均衡器，则请通过启用跨区域负载均衡来确保每个 EC2 实例获得相似的流量。这有助于限制失效转移事件期间增加的负载对现有实例的影响，并比没有跨区域负载平衡时具有更高的弹性。
- 确保正确配置了 Elastic Load Balancing 运行状况检查，并在自动扩缩组上启用了这些检查。然后，如果实例的运行状况检查失败，则 Elastic Load Balancing 会停止向其发送流量并将流量重新路由到运行状况良好的实例，而 Amazon EC2 Auto Scaling 会替换运行状况不佳的实例。

Amazon EC2 Auto Scaling 通过以下方式帮助支持您的应用程序弹性需求：

- 检查实例是否存在运行状况和可访问性问题。如果某个实例运行状况不佳，则它将自动终止该实例，并启动新实例。
- 如果动态扩展策略生效，则根据传入流量自动扩展容量。
- 检测支持扩展策略的 Amazon CloudWatch 指标的可靠性问题，并在缺乏可靠性指标（例如缺少数据点）时暂停横向缩减活动。
- 尝试在您的组扩展时在每个已启用的可用区中维持相同数量的实例。
- 使用可用区以保持高可用性。当可用区运行状况不佳时，Amazon EC2 Auto Scaling 将执行以下操作：
  - 在为自动扩缩组启用的其他可用区中启动新实例。
  - 当运行状况不佳的可用区恢复正常状态时，在所有已启用的可用区中重新分配实例。
- 如果某个实例无法在指定的可用区中启动，则继续尝试在其他已启用的可用区中启动实例。
- 自动向与自动扩缩组关联的负载均衡器注册和取消注册实例。这样，您无需单独注册和取消注册实例。

有关帮助支持 Amazon EC2 数据弹性和备份需求的功能的信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 中的弹性。

Amazon EC2 Auto Scaling 中的基础设施安全性

作为一项托管服务，Amazon EC2 Auto Scaling 由Amazon Web Services：安全流程概览白皮书中所述的Amazon全球网络安全流程提供保护。

您可以使用 Amazon 发布的 API 调用通过网络访问 Amazon EC2 Auto Scaling。客户端必须支持传输层安全性 (TLS) 1.0 或更高版本。建议使用 TLS 1.2 或更高版本。客户端还必须支持具有完全向前保密 (PFS) 的密码套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Ephemeral Diffie-Hellman (ECDHE)。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 Amazon Security Token Service (Amazon STS) 生成临时安全凭证来对请求进行签名。

您还可以将虚拟私有云（VPC）端点用于 Amazon EC2 Auto Scaling。接口 VPC 端点使 Amazon VPC 资源可以使用其私有 IP 地址访问 Amazon EC2 Auto Scaling，而无需接触公有互联网。有关更多信息，请参阅Amazon EC2 Auto Scaling 和接口 VPC 终端节点 (p. 339)。

相关主题

- 适用于 Linux 实例的 Amazon EC2 用户指南中的 Amazon EC2 中的基础设施安全性

Amazon EC2 Auto Scaling 和接口 VPC 终端节点

您可以通过将 Amazon EC2 Auto Scaling 配置为使用接口 VPC 终端节点，以改善 VPC 的安保状况。接口端点由 Amazon PrivateLink 提供支持，此技术让您能够将 VPC 和 Amazon EC2 Auto Scaling 之间的所有网络流量限制在 Amazon 网络内，从而实现对 Amazon EC2 Auto Scaling API 的私有访问。借助接口终端节点，您也不需要 Internet 网关、NAT 设备或虚拟专用网关。

不要求您配置 Amazon PrivateLink，但推荐进行配置。有关 Amazon PrivateLink 和 VPC 终端节点的更多信息，请参阅《Amazon PrivateLink 指南》中的什么是 Amazon PrivateLink？

主题

- 创建接口 VPC 终端节点 (p. 339)
- 创建 VPC 终端节点策略 (p. 339)

创建接口 VPC 终端节点

使用以下服务名称为 Amazon EC2 Auto Scaling 创建端点：

```
com.amazonaws.region.autoscaling
```

有关更多信息，请参阅《Amazon PrivateLink 指南》中的使用接口 VPC 终端节点访问 Amazon 服务。

您无需更改任何 Amazon EC2 Auto Scaling 设置。Amazon EC2 Auto Scaling 使用服务端点或私有接口 VPC 终端节点（二者中在使用中的那个）调用其他 Amazon 服务。

创建 VPC 终端节点策略

您可以向 VPC 终端节点附加策略来控制对 Amazon EC2 Auto Scaling API 的访问。该策略指定：

- 可执行操作的委托人。
- 可执行的操作。
- 可对其执行操作的资源。

以下示例显示了一个 VPC 终端节点策略，该策略拒绝所有人通过终端节点删除扩展策略的权限。示例策略还授予所有人执行所有其他操作的权限。

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
```

```
            "Action": "autoscaling:DeleteScalingPolicy",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

有关更多信息，请参阅《Amazon PrivateLink 指南》中的 VPC 终端节点策略。

Amazon EC2 Auto Scaling 问题排查

Amazon EC2 Auto Scaling 提供特定的描述性错误消息来帮助您排查问题。可以从扩展活动的描述中发现错误消息。

主题

- 检索来自扩缩活动的错误消息 (p. 341)
- 其他故障排除资源 (p. 342)
- Amazon EC2 Auto Scaling 问题排查：EC2 实例启动失败 (p. 343)
- Amazon EC2 Auto Scaling 问题排查：AMI 问题 (p. 347)
- Amazon EC2 Auto Scaling 问题排查：负载均衡器问题 (p. 348)
- Amazon EC2 Auto Scaling 问题排查：启动模板 (p. 350)
- Amazon EC2 Auto Scaling 问题排查：运行状况检查 (p. 351)

检索来自扩缩活动的错误消息

要从扩展活动的描述中检索错误消息，请使用 describe-scaling-activities 命令。您拥有可追溯到 6 周的扩展活动记录。扩展活动按开始时间排序，首先列出最新的扩展活动。

> **Note**
>
> 在 Amazon EC2 Auto Scaling 控制台中，在该 Auto Scaling 组的 Activity（活动）选项卡的活动历史记录中也会显示扩缩活动。

要查看特定 Auto Scaling 组的扩展活动，请使用以下命令。

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

在下面的示例响应中，StatusCode 包含活动的当前状态，StatusMessage 包含错误消息。

```
{
    "Activities": [
        {
            "ActivityId": "3b05dbf6-037c-b92f-133f-38275269dc0f",
            "AutoScalingGroupName": "my-asg",
            "Description": "Launching a new EC2 instance: i-003a5b3ffe1e9358e.  Status
 Reason: Instance failed to complete user's Lifecycle Action: Lifecycle Action with token
 e85eb647-4fe0-4909-b341-a6c42d8aba1f was abandoned: Lifecycle Action Completed with
 ABANDON Result",
            "Cause": "At 2021-01-11T00:35:52Z a user request created an AutoScalingGroup
 changing the desired capacity from 0 to 1.  At 2021-01-11T00:35:53Z an instance was
 started in response to a difference between desired and actual capacity, increasing the
 capacity from 0 to 1.",
            "StartTime": "2021-01-11T00:35:55.542Z",
            "EndTime": "2021-01-11T01:06:31Z",
            "StatusCode": "Cancelled",
            "StatusMessage": "Instance failed to complete user's Lifecycle Action:
 Lifecycle Action with token e85eb647-4fe0-4909-b341-a6c42d8aba1f was abandoned: Lifecycle
 Action Completed with ABANDON Result",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-
west-2b\"...}",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
        },
```

```
    ...
    ]
}
```

有关输出中字段的描述，请参阅 Amazon EC2 Auto Scaling API 参考中的活动。

要查看已删除组的扩展活动

要在删除自动扩缩组后查看扩展活动，请将 `--include-deleted-groups` 选项添加到以下 describe-scaling-activities 命令。

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg --include-
deleted-groups
```

以下是示例响应，其中包含已删除组的扩展活动。

```
{
    "Activities": [
        {
            "ActivityId": "e1f5de0e-f93e-1417-34ac-092a76fba220",
            "AutoScalingGroupName": "my-asg",
            "Description": "Launching a new EC2 instance.  Status Reason: Your Spot request
 price of 0.001 is lower than the minimum required Spot request fulfillment price of
 0.0031. Launching EC2 instance failed.",
            "Cause": "At 2021-01-13T20:47:24Z a user request update of AutoScalingGroup
 constraints to min: 1, max: 5, desired: 3 changing the desired capacity from 0 to 3.  At
 2021-01-13T20:47:27Z an instance was started in response to a difference between desired
 and actual capacity, increasing the capacity from 0 to 3.",
            "StartTime": "2021-01-13T20:47:30.094Z",
            "EndTime": "2021-01-13T20:47:30Z",
            "StatusCode": "Failed",
            "StatusMessage": "Your Spot request price of 0.001 is lower than the minimum
 required Spot request fulfillment price of 0.0031. Launching EC2 instance failed.",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-
west-2b\"...}",
            "AutoScalingGroupState": "Deleted",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
        },
     ...
    ]
}
```

其他故障排除资源

以下页面提供了有关对 Amazon EC2 Auto Scaling 问题进行故障排除的其他信息。

- 验证 Auto Scaling 组的扩缩活动 (p. 160)
- 在 Amazon EC2 Auto Scaling 控制台中查看监控图表 (p. 254)
- Auto Scaling 实例的运行状况检查 (p. 241)
- 生命周期钩子的注意事项和限制 (p. 191)
- 完成生命周期操作 (p. 201)
- 使用 Amazon VPC 为 Auto Scaling 实例提供网络连接 (p. 300)
- 临时从 Auto Scaling 组中移除实例 (p. 232)
- 禁用 Auto Scaling 组的扩缩策略 (p. 161)

- 暂停和恢复 Auto Scaling 组的进程 (p. 236)
- 控制在横向缩减过程中要终止的 Auto Scaling 实例 (p. 222)
- 删除 Auto Scaling 基础设施 (p. 117)
- Amazon EC2 Auto Scaling 的配额 (p. 10)

以下 Amazon 资源也会有所帮助：

- Amazon 知识中心中的 Amazon EC2 Auto Scaling 主题
- Amazon re:Post 上的 Amazon EC2 Auto Scaling 问题
- Amazon 计算博客中的 Amazon EC2 Auto Scaling 帖子
- Amazon CloudFormation 用户指南中的 CloudFormation 故障排除

故障排除通常需要由专家或多个帮助者进行迭代查询和发现。如果尝试这一部分的建议后仍然存在问题，请联系 Amazon Web Services Support [在 Amazon Web Services Management Console 中，单击 Support（支持）、Support Center（支持中心）] 或使用 Amazon EC2 Auto Scaling 标签在 Amazon re:Post 上提问。

Amazon EC2 Auto Scaling 问题排查：EC2 实例启动失败

此页提供有关未能启动的 EC2 实例的信息、可能的原因，以及可用来解决问题的步骤。

要检索错误消息，请参阅检索来自扩缩活动的错误消息 (p. 341)。

EC2 实例启动失败时，您可能会收到一条或多条以下错误消息：

启动问题

- 当前不支持请求的配置。 (p. 343)
- 安全组 <该安全组的名称> 不存在。启动 EC2 实例失败。 (p. 344)
- 密钥对 <与您的 EC2 实例相关联的密钥对> 不存在。启动 EC2 实例失败。 (p. 344)
- 不再支持请求的可用区。请重新尝试请求... (p. 344)
- 请求的实例类型（<实例类型>）在请求的可用区（<实例可用区>）中不受支持... (p. 344)
- 您的竞价请求价格 0.015 低于要求的最低竞价请求履行价格 0.0735... (p. 345)
- 设备名称 <device name> 无效/设备名称上传无效。启动 EC2 实例失败。 (p. 345)
- 用于参数 virtualName 的值（<与实例存储设备相关联的名称>）无效... (p. 345)
- 实例存储 AMI 不支持 EBS 块储存设备映射。 (p. 345)
- 置放群组可能无法与类型为“m1.large”的实例一起使用。启动 EC2 实例失败。 (p. 346)
- Client.InternalError：客户端在启动时出错。 (p. 346)
- 我们目前在您请求的可用区中没有足够的 <实例类型> 容量。启动 EC2 实例失败。 (p. 346)
- 没有与您的请求匹配的竞价容量。启动 EC2 实例失败。 (p. 347)
- 已运行 <实例数量> 个实例。启动 EC2 实例失败。 (p. 347)

当前不支持请求的配置。

- 原因：启动模板或启动配置中的某些选项可能与实例类型不兼容，或者实例配置可能在您请求的 Amazon 区域或可用区中不受支持。
- 解决方案：

尝试其他实例配置。要搜索符合您要求的实例类型，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的查找 Amazon EC2 实例类型。

有关解决该问题的更多指导，请检查以下内容：

- 确保您选择了实例类型支持的 AMI。例如，如果实例类型使用基于 ARM 的 Amazon 引力处理器而不是 Intel Xeon 处理器，您需要一个 ARM 兼容的 AMI。
- 测试实例类型在所请求的区域和可用区中可用。最新一代实例类型可能尚未在给定区域或可用区域中可用。旧一代实例类型可能在给定区域或可用区中不可用。要搜索按位置（区域或可用区）提供的实例类型，可使用 describe-instance-type-offerings 命令。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的查找 Amazon EC2 实例类型。
- 如果您使用专用实例或专用主机，请确保选择了作为专用实例或专用主机受支持的实例类型。

安全组 <该安全组的名称> 不存在。启动 EC2 实例失败。

- 原因：可能已删除启动模板或启动配置中指定的安全组。
- 解决方案：
  1. 使用 describe-security-groups 命令获取与您的账户相关联的安全组列表。
  2. 从该列表中选择要使用的安全组。要改为创建安全组，请使用 create-security-group 命令。
  3. 创建新的启动模板或启动配置。
  4. 使用 update-auto-scaling-group 命令通过新的启动模板或启动配置更新您的 Auto Scaling 组。

密钥对 <与您的 EC2 实例相关联的密钥对> 不存在。启动 EC2 实例失败。

- 原因：可能已删除启动实例时使用的密钥对。
- 解决方案：
  1. 使用 describe-key-pairs 命令获取可用密钥对的列表。
  2. 从该列表中选择要使用的密钥对。要改为创建密钥对，请使用 create-key-pair 命令。
  3. 创建新的启动模板或启动配置。
  4. 使用 update-auto-scaling-group 命令通过新的启动模板或启动配置更新您的 Auto Scaling 组。

不再支持请求的可用区。请重新尝试请求...

- 错误消息：不再支持请求的可用区。请通过不指定可用区或选择 <提供的可用区的列表> 来重新尝试请求。启动 EC2 实例失败。
- 原因：当前可能无法使用与 Auto Scaling 组相关联的可用区。
- 解决方案：采用错误消息中的建议更新 Auto Scaling 组。

请求的实例类型（<实例类型>）在请求的可用区（<实例可用区>）中不受支持...

- 错误消息：您请求的实例类型（<实例类型>）在请求的可用区（<实例可用区>）中不受支持。请通过不指定可用区或选择 <支持该实例类型的可用区列表> 来重新尝试请求。启动 EC2 实例失败。

- 原因：当前可能无法在 Auto Scaling 组中指定的可用区内使用您选择的实例类型。
- 解决方案：采用错误消息中的建议更新 Auto Scaling 组。

您的竞价请求价格 0.015 低于要求的最低竞价请求履行价格 0.0735...

- 原因：请求中的 Spot 最高价低于选定的实例类型的 Spot 价格。
- 解决方案：提交具有较高 Spot 最高价（可能是按需价格）的新请求。之前，您支付的 Spot 价格是基于出价的。今天，您支付当前 Spot 价格。通过设置更高的最高价，Amazon EC2 Spot 服务可以更好地启动和保持您所需的容量。

设备名称 <device name> 无效/设备名称上传无效。启动 EC2 实例失败。

- 原因 1：启动模板或启动配置中的块储存设备映射所包含的块储存设备名称可能无法使用或目前不受支持。
- 解决方案：
  1. 验证哪些设备名称可用于您的特定实例配置。有关设备命名的信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 Linux 实例上的设备命名。
  2. 手动创建不是 Auto Scaling 组一部分的 Amazon EC2 实例并调查问题。如果块储存设备命名配置与 Amazon Machine Image (AMI) 中的名称冲突，则实例将在启动期间失败。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的块储存设备映射。
  3. 确认您的实例已成功启动后，使用 describe-volumes 命令查看如何向实例公开卷。
  4. 使用卷描述中列出的设备名称创建新的启动模板或启动配置。
  5. 使用 update-auto-scaling-group 命令通过新的启动模板或启动配置更新您的 Auto Scaling 组。

用于参数 virtualName 的值（<与实例存储设备相关联的名称>）无效...

- 错误消息：virtualName 参数的值（<与实例存储设备相关联的名称>）无效。预期格式："ephemeralNUMBER"。启动 EC2 实例失败。
- 原因：与块储存设备相关联的虚拟名称的指定格式不正确。
- 解决方案：
  1. 通过在 `virtualName` 参数中指定设备名称创建新的启动模板或启动配置。有关设备名称格式的信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 Linux 实例上的设备命名。
  2. 使用 update-auto-scaling-group 命令通过新的启动模板或启动配置更新您的 Auto Scaling 组。

实例存储 AMI 不支持 EBS 块储存设备映射。

- 原因：启动模板或启动配置中指定的块储存设备映射在您的实例上不受支持。
- 解决方案：
  1. 使用实例类型支持的块储存设备映射来创建新的启动模板或启动配置。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的块储存设备映射。
  2. 使用 update-auto-scaling-group 命令通过新的启动模板或启动配置更新您的 Auto Scaling 组。

置放群组可能无法与类型为“m1.large”的实例一起使用。启动 EC2 实例失败。

- 原因：您的集群置放群组包含无效实例类型。
- 解决方案：
  1. 有关置放群组支持的有效实例类型的信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的置放群组。
  2. 按照置放群组中的详细说明创建新的置放群组。
  3. 或者，也可以使用受支持的实例类型创建新的启动模板或启动配置。
  4. 使用 update-auto-scaling-group 命令通过新的置放群组、启动模板或启动配置更新您的 Auto Scaling 组。

Client.InternalError：客户端在启动时出错。

- 问题：Amazon EC2 Auto Scaling 尝试启动具有加密 EBS 卷的实例，但服务相关角色没有权限访问用于对其进行加密的 Amazon KMS 客户托管密钥。有关更多信息，请参阅与加密卷结合使用时必需的 Amazon KMS 密钥策略 (p. 333)。
- 原因 1：您需要密钥策略，该策略向适当的服务相关角色授予使用客户托管密钥的权限。
- 解决方案 1：允许服务相关角色使用客户管理密钥，如下所示：
  1. 确定将哪个服务相关角色用于此 Auto Scaling 组。
  2. 更新客户托管密钥上的密钥策略并允许服务相关角色使用客户托管密钥。
  3. 更新 Auto Scaling 组来使用服务相关角色。

  有关允许服务相关角色使用客户托管密钥的密钥策略示例，请参阅 示例 1：允许访问客户托管密钥的关键策略部分 (p. 335)。
- 原因 2：如果客户托管密钥和 Auto Scaling 组在不同的 Amazon 账户中，您需要配置对客户托管密钥的跨账户访问权限，以便向适当的服务相关角色授予使用客户托管密钥的权限。
- 解决方案 2：允许外部账户中的服务相关角色使用本地账户中的客户托管密钥，如下所示：
  1. 更新客户托管密钥上的密钥策略，以允许 Auto Scaling 组账户访问客户托管密钥。
  2. 在 Auto Scaling 组账户中定义可以创建授权的 IAM 用户或角色。
  3. 确定将哪个服务相关角色用于此 Auto Scaling 组。
  4. 使用适当的服务相关角色作为被授权者委托人，创建对客户托管密钥的授权。
  5. 更新 Auto Scaling 组来使用服务相关角色。

  有关更多信息，请参阅示例 2：允许跨账户访问客户托管密钥的关键策略部分 (p. 335)。
- 解决方案 3：将同一 Amazon 账户中的客户托管密钥作为 Auto Scaling 组。
  1. 使用属于与 Auto Scaling 组相同的账户中的另一个客户托管密钥，复制并重新加密快照。
  2. 允许服务相关角色使用新的客户托管密钥。请参阅解决方案 1 的步骤。

我们目前在您请求的可用区中没有足够的 <实例类型> 容量。启动 EC2 实例失败。

- 错误消息：您请求的可用区（<请求的可用区>）中当前没有足够的 <实例类型> 容量。我们的系统将调配额外的容量。您当前可以通过不在请求中指定可用区，或者选择 <当前支持该实例类型的可用区列表> 来获取 <实例类型> 容量。启动 EC2 实例失败。
- 原因：现在，Amazon EC2 不能在请求的可用区中支持您的实例类型。

- 解决方案：

  要解决该问题，请尝试以下操作：

  - 等待几分钟，然后再次提交您的请求；容量可能经常转移。
  - 按照错误消息中的建议提交新请求。
  - 提交减少了实例数（可在后期阶段增加）的新请求。

没有与您的请求匹配的竞价容量。启动 EC2 实例失败。

- 原因：目前没有足够的备用容量来满足您的竞价型实例请求。
- 解决方案：

  要解决该问题，请尝试以下操作：

  - 等待几分钟；容量可能经常转移。竞价请求会继续自动发起启动请求，直到有可用容量为止。当容量变为可用时，Amazon EC2 竞价服务将满足竞价请求。
  - 请遵循使用不同实例类型集的最佳实践，以便您不依赖于某一特定实例类型。有关更多信息，包括成功使用竞价型实例的最佳实践列表，请参阅 Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。
  - 提交减少了实例数（可在后期阶段增加）的新请求。

已运行 <实例数量> 个实例。启动 EC2 实例失败。

- 原因：您已达到可在某一区域中启动的实例数限制。在创建 Amazon 账户时，我们根据区域设置可运行的实例数的默认限制。
- 解决方案：

  要解决该问题，请尝试以下操作：

  - 如果您当前的限制不足以满足需求，您可以根据区域请求提高配额。有关更多信息，请参见适用于 Linux 实例的 Amazon EC2 用户指南中的 Amazon EC2 服务配额。
  - 提交减少了实例数（可在后期阶段增加）的新请求。

Amazon EC2 Auto Scaling 问题排查：AMI 问题

本页提供与 AMI 相关联的问题的有关信息、可能原因，以及可用来解决这些问题的步骤。

要检索错误消息，请参阅检索来自扩缩活动的错误消息 (p. 341)。

当 AMI 的问题导致 EC2 实例启动失败时，您可能会收到一条或多条以下错误消息：

AMI 问题

- AMI ID <您的 AMI 的 ID> 不存在。启动 EC2 实例失败。 (p. 348)
- AMI <AMI ID> 正在等待，无法运行。启动 EC2 实例失败。 (p. 348)
- 用于参数 virtualName 的值 (<ami ID>) 无效。 (p. 348)
- 请求的实例类型的架构 (i386) 与用于 ami-6622f00f (x86_64) 的清单中的架构不匹配。启动 EC2 实例失败。 (p. 348)

AMI ID <您的 AMI 的 ID> 不存在。启动 EC2 实例失败。

- 原因：创建启动模板或启动配置后，可能已删除 AMI。
- 解决方案：
  1. 使用有效的 AMI 创建新的启动模板或启动配置。
  2. 使用 update-auto-scaling-group 命令通过新的启动模板或启动配置更新您的 Auto Scaling 组。

AMI <AMI ID> 正在等待，无法运行。启动 EC2 实例失败。

- 原因：您可能刚创建 AMI（通过获取运行实例的快照或任何其他方式），它可能还无法使用。
- 解决方案：必须等待您的 AMI 可用后，才能创建启动模板或启动配置。

用于参数 virtualName 的值 (<ami ID>) 无效。

- 原因：不正确的值。`virtualName` 参数表示与设备相关联的虚拟名称。
- 解决方案：
  1. 通过指定使用 `virtualName` 参数的实例的虚拟设备名称来创建新的启动模板或启动配置。
  2. 使用 update-auto-scaling-group 命令通过新的启动模板或启动配置更新您的 Auto Scaling 组。

请求的实例类型的架构 (i386) 与用于 ami-6622f00f (x86_64) 的清单中的架构不匹配。启动 EC2 实例失败。

- 原因：启动模板或启动配置中提到的 `InstanceType` 的架构与映像架构不匹配。
- 解决方案：
  1. 使用与请求的实例类型的架构相匹配的 AMI 架构来创建新的启动模板或启动配置。
  2. 使用 update-auto-scaling-group 命令通过新的启动模板或启动配置更新您的 Auto Scaling 组。

Amazon EC2 Auto Scaling 问题排查：负载均衡器问题

本页提供与 Auto Scaling 组相关联的负载均衡器所导致的问题有关的信息、可能原因，以及可用来解决这些问题的步骤。

要检索错误消息，请参阅检索来自扩缩活动的错误消息 (p. 341)。

如果与 Auto Scaling 组关联的负载均衡器导致 EC2 实例启动失败，您可能会收到一条或多条以下错误消息：

负载均衡器问题

- 一个或多个目标组未找到。验证负载均衡器配置失败。 (p. 349)

- 找不到负载均衡器 <your load balancer>。验证负载均衡器配置失败。 (p. 349)
- 名为 <负载均衡器名称> 的活动负载均衡器不存在。更新负载均衡器配置失败。 (p. 349)
- EC2 实例 <实例 ID> 不在 VPC 内。更新负载均衡器配置失败。 (p. 349)
- EC2 实例 <实例 ID> 在 VPC 中。更新负载均衡器配置失败。 (p. 350)

一个或多个目标组未找到。验证负载均衡器配置失败。

问题：当您的自动扩缩组启动实例时，Amazon EC2 Auto Scaling 会尝试验证与自动扩缩组关联的 Elastic Load Balancing 资源是否存在。当找不到目标组时，扩展活动会失败，您会收到 `One or more target groups not found. Validating load balancer configuration failed.` 错误消息。

原因 1：关联到您的自动扩缩组的目标组已被删除。

解决方案 1：可以创建无目标组的新自动扩缩组，也可以使用 Amazon EC2 Auto Scaling 控制台或 detach-load-balancer-target-groups 命令从自动扩缩组中删除未使用的目标组。

原因 2：目标组存在，但是创建自动扩缩组时，在尝试指定目标组 ARN 时出现问题。资源的创建顺序不正确。

解决方案 2：创建新的自动扩缩组，并在最后指定目标组名称。

找不到负载均衡器 <your load balancer>。验证负载均衡器配置失败。

问题：当您的自动扩缩组启动实例时，Amazon EC2 Auto Scaling 会尝试验证与自动扩缩组关联的 Elastic Load Balancing 资源是否存在。当找不到经典负载均衡器时，扩展活动会失败，您会收到 `Cannot find Load Balancer <your load balancer>. Validating load balancer configuration failed.` 错误消息。

原因 1：已删除经典负载均衡器。

解决方案 1：可以创建无负载均衡器的新自动扩缩组，也可以使用 Amazon EC2 Auto Scaling 控制台或 detach-load-balancers 命令从自动扩缩组中删除未使用的负载均衡器。

原因 2：经典负载均衡器存在，但是创建自动扩缩组时，在尝试指定负载均衡器名称时出现问题。资源的创建顺序不正确。

解决方案 2：创建新的 Auto Scaling 组，并在最后指定负载均衡器名称。

名为 <负载均衡器名称> 的活动负载均衡器不存在。更新负载均衡器配置失败。

原因：可能已删除指定的负载均衡器。

解决方案：可以创建新的负载均衡器，然后创建新的 Auto Scaling 组，也可以创建无负载均衡器的新 Auto Scaling 组。

EC2 实例 <实例 ID> 不在 VPC 内。更新负载均衡器配置失败。

原因：VPC 中不存在指定的实例。

解决方案：可以删除与实例相关联的负载均衡器，或者创建新的 Auto Scaling 组。

EC2 实例 <实例 ID> 在 VPC 中。更新负载均衡器配置失败。

原因：负载均衡器在 EC2-Classic 中，但 Auto Scaling 组在 VPC 中。

解决方案：确保负载均衡器和 Auto Scaling 组在同一网络（EC2-Classic 或 VPC）中。

Amazon EC2 Auto Scaling 问题排查：启动模板

使用以下信息可帮助您诊断和修复在尝试为自动扩缩组指定启动模板时可能遇到的常见问题。

无法启动实例

如果您无法使用已指定的启动模板启动任何实例，请检查以下一般故障排除内容：Amazon EC2 Auto Scaling 问题排查：EC2 实例启动失败 (p. 343)。

您必须使用有效的完整启动模板（无效值）

问题：当您尝试为某个自动扩缩组指定启动模板时，收到 `You must use a valid fully-formed launch template` 错误。您可能遇到此错误，因为只有在使用启动模板的自动扩缩组被创建或更新时，才会对启动模板中的值进行验证。

原因 1：如果您收到 `You must use a valid fully-formed launch template` 错误，则表示有些问题导致 Amazon EC2 Auto Scaling 将启动模板的某些内容视为无效。这是一个通用错误，可能由几种不同的原因导致。

解决方案 1：请尝试以下步骤进行故障排除：

1. 请注意错误消息的第二部分以查找更多信息。在遇到 `You must use a valid fully-formed launch template` 错误后，请参阅更具体的错误消息，该消息标识您需要解决的问题。
2. 如果您无法找到原因，请使用 run-instances 命令测试您的启动模板。使用 `--dry-run` 选项，如以下示例中所示。这使您可以重现问题并提供有关问题原因的见解。

```
aws ec2 run-instances --launch-template LaunchTemplateName=my-template,Version='1' --
dry-run
```

3. 如果值无效，请验证指定的资源是否存在且正确。例如，在您指定 Amazon EC2 密钥对时，资源必须存在于您从中创建或更新自动扩缩组的账户和区域中。
4. 如果缺少预期信息，请验证您的设置并根据需要调整启动模板。
5. 进行更改后，使用 `--dry-run` 选项重新运行 run-instances 命令，以验证您的启动模板是否使用有效的值。

有关更多信息，请参阅为 Auto Scaling 组创建启动模板 (p. 20)。

您没有权限使用启动模板（权限不足）

问题：当您尝试为某个自动扩缩组指定启动模板时，收到 `You are not authorized to use launch template` 错误。

原因 1：如果您尝试使用启动模板，但您使用的 IAM 凭证没有足够的权限，则会收到一条错误，显示您没有权限使用启动模板。

解决方案 1：验证您用于发出请求的 IAM 凭证具有调用所需 EC2 API 操作的权限，包括 `ec2:RunInstances` 操作。如果在启动模板中指定了任何标签，您还必须拥有使用 `ec2:CreateTags` 操作的权限。

解决方案 2：验证您用于发出请求的 IAM 凭证被分配了 `AmazonEC2FullAccess` 策略。该 Amazon 托管策略授予对所有 Amazon EC2 资源和相关服务的完全访问权限，包括 Amazon EC2 Auto Scaling、CloudWatch 和 Elastic Load Balancing。

原因 2：如果您尝试使用指定实例配置文件的启动模板，则必须具有传递与实例配置文件关联的 IAM 角色的 IAM 权限。

解决方案 3：验证您用于发出请求的 IAM 凭证具有将指定角色传递给 Amazon EC2 Auto Scaling 服务的正确 `iam:PassRole` 权限。有关更多信息和示例 IAM policy，请参阅 在 Amazon EC2 实例上运行的应用程序的 IAM 角色 (p. 331)。有关实例配置文件相关的进一步故障排查主题，请参阅 IAM 用户指南中的 Amazon EC2 和 IAM 故障排查。

原因 3：如果您尝试使用在另一个 Amazon Web Services 账户 中指定 AMI 的启动模板，且该 AMI 是私有的，不与您正在使用的 Amazon Web Services 账户 共享，则您会收到一条错误，显示您没有权限使用该启动模板。

解决方案 4：验证 AMI 的权限是否包括您正在使用的账户。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的与特定 Amazon Web Services 账户 共享 AMI。

> **Important**
>
> 有关启动模板的权限的更多信息，包括示例 IAM policy，请参阅 启动模板支持 (p. 328)。

Amazon EC2 Auto Scaling 问题排查：运行状况检查

此页面提供有关由于运行状况检查而终止的 EC2 实例信息。它描述潜在原因以及可用于解决问题的步骤。

要检索错误消息，请参阅检索来自扩缩活动的错误消息 (p. 341)。

运行状况检查问题

- 实例因 EC2 实例状态检查失败而停止使用 (p. 351)
- 实例因 EC2 计划重新启动而停止使用 (p. 352)
- 实例因 EC2 运行状况检查而停止服务指示它已终止或停止 (p. 352)
- 实例因 ELB 系统运行状况检查失败而停止服务 (p. 353)

> **Note**
>
> 当 Amazon EC2 Auto Scaling 终止 Auto Scaling 组中的实例时，包括实例终止的原因不是由于扩展活动而导致的情况下，您会收到通知。有关更多信息，请参阅在 Auto Scaling 组扩展时获取 Amazon SNS 通知 (p. 258)。
> 以下各节介绍了最常见的运行状况检查错误和您将遇到的原因。如果您有其他问题，请参阅以下 Amazon 知识中心文章以获取其他故障排查帮助：
>
> - Amazon EC2 Auto Scaling 为何终止实例？
> - Amazon EC2 Auto Scaling 为何终止运行状况不佳的实例？

实例因 EC2 实例状态检查失败而停止使用

问题：Auto Scaling 实例未通过 Amazon EC2 状态检查。

原因 1：如果存在导致 Amazon EC2 考虑您的 Auto Scaling 组中的实例受损的问题，Amazon EC2 Auto Scaling 会自动替换受损的实例，作为其运行状况检查的一部分。状态检查是内置到 Amazon EC2 中的，所以不能禁用或删除。当实例状态检查失败时，通常必须通过更改实例配置来自行解决问题直到应用程序不再出现任何问题。

解决方案 1：要解决该问题，请按照以下步骤操作：

1. 手动创建不是 Auto Scaling 组一部分的 Amazon EC2 实例并调查问题。有关调查受损实例的一般帮助，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的通过故障状态检查来排查实例问题和适用于 Windows 实例的 Amazon EC2 用户指南中的对 Windows 实例进行故障排查。
2. 在确认实例已成功启动且运行状况良好后，请将新的、无错误的实例配置部署到 Auto Scaling 组。
3. 删除您创建的实例以免对 Amazon 账户继续产生费用。

原因 2：运行状况检查宽限期与实例启动时间不匹配。

解决方案 2：将您的 Auto Scaling 组的运行状况检查宽限期编辑为适合您的应用程序的适当时间段。在 Auto Scaling 组中启动的实例需要足够的预热时间（宽限期），以防因运行状况检查替换而提前终止。有关更多信息，请参阅设置自动扩缩组的运行状况检查宽限期 (p. 246)。

实例因 EC2 计划重新启动而停止使用

问题：当计划事件指示实例存在问题时，Auto Scaling 实例被替换。

原因：Amazon EC2 Auto Scaling 将利用未来的计划维护或停用事件替换实例。

解决方案：这些事件不会频繁发生。如果您需要在正在终止的实例上或正在启动的实例上执行某些操作，则可以使用生命周期钩子。这些挂钩允许您在 Amazon EC2 Auto Scaling 启动或终止实例时执行自定义操作。有关更多信息，请参阅Amazon EC2 Auto Scaling 生命周期钩子 (p. 191)。

如果您不想因计划事件而替换实例，可以暂停 Auto Scaling 组的运行状况检查流程。有关更多信息，请参阅暂停和恢复 Auto Scaling 组的进程 (p. 236)。

实例因 EC2 运行状况检查而停止服务指示它已终止或停止

问题：已停止、重新启动或终止的 Auto Scaling 实例将被替换。

原因 1：用户已手动停止、重新启动或终止实例。

解决方案 1：如果运行状况检查因用户手动停止、重新启动或终止实例而失败，则这是由于 Amazon EC2 Auto Scaling 运行状况检查的工作原理造成的。实例必须运行正常且可访问。如果您需要重启 Auto Scaling 组中的实例，我们建议您首先将实例置于备用状态。有关更多信息，请参阅临时从 Auto Scaling 组中移除实例 (p. 232)。

请注意，当您手动终止实例时，必须在实际终止实例之前完成终止生命周期挂钩和 Elastic Load Balancing 取消注册（以及连接耗尽）。

原因 2：Amazon EC2 Auto Scaling 尝试在 Amazon EC2 竞价服务中断实例后替换竞价型实例，因为 Spot 价格上涨超过您的最高价或容量不再可用。

解决方案 2：无法保证在任何给定时间点存在竞价型实例来满足请求。但是，您可以尝试以下操作：

- 使用更高的 Spot 最高价（可能是按需价格）。通过设置更高的最高价，Amazon EC2 竞价服务可以更好地启动和保持您所需的容量。
- 通过在多个可用区中运行多个实例类型，增加您可以从中启动实例的不同容量池的数量。有关更多信息，请参阅 Auto Scaling 组具有多个实例类型和购买选项 (p. 50)。

- 如果您使用多个实例类型，请考虑启用容量再平衡功能。如果您希望 Amazon EC2 竞价服务在运行实例终止之前尝试启动新的竞价型实例，这将非常有用。有关更多信息，请参阅使用容量重新平衡来处理 Amazon EC2 竞价型实例中断 (p. 262)。

实例因 ELB 系统运行状况检查失败而停止服务

问题：Auto Scaling 实例可能会通过 EC2 状态检查。但是，它们可能无法对已注册 Auto Scaling 组的目标组或经典负载均衡器进行 Elastic Load Balancing 运行状况检查。

原因：如果您的 Auto Scaling 组依赖于 Elastic Load Balancing 提供的运行状况检查，Amazon EC2 Auto Scaling 通过检查 EC2 状态检查和 Elastic Load Balancing 运行状况检查的结果来确定实例的运行状况。负载均衡器通过向每个实例发送请求并等待正确响应或与实例建立连接来执行运行状况检查。实例未能通过 Elastic Load Balancing 运行状况检查，可能是因为实例中运行的应用程序发生问题，导致负载均衡器将实例视为停止服务。有关更多信息，请参阅向 Auto Scaling 组添加 Elastic Load Balancing 运行状况检查 (p. 276)。

解决方案 1：要通过 Elastic Load Balancing 运行状况检查，请执行以下操作：

- 记住负载均衡器所需的成功代码，并且验证应用程序已正确配置为在成功时返回这些代码。
- 验证负载均衡器和 Auto Scaling 组的安全组是否已正确配置。
- 验证目标组的运行状况检查设置是否已正确配置。您定义每个目标组的负载均衡器的运行状况检查设置。
- 考虑向 Auto Scaling 组添加启动生命周期挂钩，以确保实例上的应用程序在生命周期挂钩结束时准备好接受流量。
- 将 Auto Scaling 组的运行状况检查宽限期设置为足够长的时间段，以支持 Elastic Load Balancing 认为新启动的实例正常运行状况检查所需的连续成功次数。
- 验证负载均衡器是否配置在与 Auto Scaling 组相同的可用区中。

有关更多信息，请参阅以下主题：

- Application Load Balancer 的用户指南中的目标组的运行状况检查
- 《Network Load Balancer 用户指南》中的目标组的运行状况检查
- 网关负载均衡器的用户指南中的目标组的运行状况检查
- 《经典负载均衡器用户指南》中的为您的经典负载均衡器配置运行状况检查

解决方案 2：更新 Auto Scaling 组以禁用 Elastic Load Balancing 运行状况检查。

Amazon EC2 Auto Scaling 资源

下列相关资源在您使用此服务的过程中会有所帮助。

- 在中国开始使用 Amazon 中的 Amazon EC2 Auto Scaling - 中国的 Amazon EC2 Auto Scaling 文档基于公有 Amazon 文档中。此文档中描述的服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅在中国开始使用 Amazon 的“Amazon EC2 Auto Scaling”部分。
- Amazon EC2：如何连接到 Linux 实例 – 了解如何连接到您启动的 Linux 实例。
- Amazon EC2：如何连接到您的 Windows 实例 – 了解如何连接到您启动的 Windows 实例。
- CloudWatch：创建账单告警以监控估算 Amazon 费用 - 了解如何使用 CloudWatch 监控预估费用。

以下附加资源可帮助您了解有关 Amazon Web Services 的更多信息。

- 入门资源中心 – 了解如何设置 Amazon Web Services 账户、加入 Amazon 社区和启动您的第一个应用程序。
- Amazon Web Services Support 中心 – 用于创建和管理 Amazon Web Services Support 案例的中心。还提供指向其他有用资源的链接，如论坛、技术常见问题、服务运行状况以及Amazon Trusted Advisor。
- Amazon Web Services Support – 提供有关 Amazon Web Services Support 的信息的主要网页，这是一个一对一的快速响应支持渠道，可以帮助您在云中构建和运行应用程序。
- 联系我们 – 用于查询有关Amazon账单、账户、事件、滥用和其他问题的中央联系点。
- Amazon 网站条款 – 有关我们的版权和商标、您的账户、许可、网站访问和其他主题的详细信息。

文档历史记录

下表介绍了自 2018 年 7 月以来对 Amazon EC2 Auto Scaling 文档的重要补充。如需对此文档更新的通知，您可以订阅 RSS 源。

变更	说明	日期
新的实例刷新功能 (p. 355)	现在，您可以选择终止或忽略待机实例并替换或忽略横向缩减保护实例，而不必等待它们变为可替换实例。您也可以回滚因实例刷新失败而产生的更改。作为本次更新的一部分，文档已扩展到包括回滚实例刷新、取消实例刷新以及了解实例刷新可配置参数的默认值等主题。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的基于实例刷新替换 Auto Scaling 实例。	2023 年 2 月 10 日
支持对 AMI ID 使用 Amazon Systems Manager 参数 (p. 355)	您现在可以在启动模板中使用 Systems Manager 参数代替 AMI ID。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的在启动模板中使用 Amazon Systems Manager 参数代替 AMI ID。	2023 年 1 月 19 日
预测性扩展建议 (p. 355)	您现在可以从 Amazon EC2 Auto Scaling 控制台获得有关评估和选择正确预测性扩展策略的建议。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的评估预测性扩展策略。	2023 年 1 月 18 日
预测式扩展预测 (p. 355)	预测式扩展生成的预测现在每六小时更新一次，而不是每天更新一次。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 的预测性扩展。	2023 年 1 月 6 日
支持 CloudWatch 指标数学 (p. 355)	创建目标跟踪扩展策略时，您现在可使用指标数学。借助指标数学，您可以查询多个 CloudWatch 指标并使用数学表达式来创建基于这些指标的新时间序列。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的使用指标数学为 Amazon EC2 Auto Scaling 创建目标跟踪策略。	2022 年 12 月 8 日
IAM 服务相关角色权限更新 (p. 355)	`AutoScalingServiceRolePolicy` 策略现在会向 Amazon EC2 Auto Scaling 授予额外的权限。有关更	2022 年 12 月 6 日

	多信息，请参阅《Amazon EC2 Auto Scaling 用户指南中的适用于 Amazon EC2 Auto Scaling 的 Amazon 托管式策略。	
新的竞价型分配策略 (p. 355)	您现在可以使用价格和容量优化分配策略从中断可能性最小、价格尽可能最低的竞价型池中请求竞价型实例。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的分配策略。	2022 年 11 月 10 日
在亚太地区（雅加达）区域支持预测性扩缩 (p. 355)	您现在可以在亚太地区（雅加达）区域创建预测性扩缩策略。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 的预测性扩展。	2022 年 10 月 13 日
支持在控股台中将自定义指标用于预测性扩缩 (p. 355)	您现在可以在从 Amazon EC2 Auto Scaling 控制台创建预测性扩缩策略时使用自定义指标。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 的预测性扩展。	2022 年 10 月 13 日
对预测性扩缩指标的 CloudWatch 监控 (p. 355)	现在，您可以使用 CloudWatch 访问监控数据以进行预测性扩缩。从而可以使用指标数学来创建可显示预测数据准确性的新时间序列。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的使用 CloudWatch 监控预测性扩缩指标。	2022 年 7 月 7 日
在亚太地区（大阪）区域支持预测性扩缩 (p. 355)	您现在可以在亚太地区（大阪）区域创建预测性扩缩策略。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 的预测性扩展。	2022 年 7 月 6 日
在更多区域支持暖池休眠 (p. 355)	您现在可以在另外四个区域的暖池中进行实例休眠：非洲（开普敦）、亚太地区（雅加达）、亚太地区（大阪）和欧洲地区（米兰）。有关暖池的更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的 Amazon EC2 Auto Scaling 暖池。	2022 年 7 月 5 日

运行状况检查更新 (p. 355)	在执行运行状况检查时，Amazon EC2 Auto Scaling 现在可以帮助您尽可能减少可能因临时问题或运行状况检查配置错误导致的停机时间。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的 Amazon EC2 Auto Scaling 如何尽可能减少停机时间。	2022 年 5 月 21 日
原定设置实例预热 (p. 355)	您现在可以统一 Auto Scaling 组的所有预热和冷却设置，并通过启用原定设置实例预热来优化持续扩展的扩缩策略的性能。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的设置 Auto Scaling 组的原定设置实例预热。	2022 年 4 月 19 日
指南更改 (p. 355)	本指南增加了一个有关与其他 Amazon 服务集成的新章节。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南中的与 Amazon EC2 Auto Scaling 集成的 Amazon 服务。	2022 年 3 月 29 日
IAM 服务相关角色权限更新 (p. 355)	`AutoScalingServiceRolePolicy` 策略现在会向 Amazon EC2 Auto Scaling 授予额外的读取权限。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南中的适用于 Amazon EC2 Auto Scaling 的 Amazon 托管式策略。	2022 年 3 月 28 日
实例元数据提供目标生命周期状态 (p. 355)	您可以从实例元数据检索 Auto Scaling 实例的目标生命周期状态。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的通过实例元数据检索目标生命周期状态。	2022 年 3 月 24 日
对新的暖池功能的支持 (p. 355)	现在，您可以将暖池中的实例休眠以停止实例，而无需删除其内存内容 (RAM)。现在，您还可以在横向缩减时使实例返回到暖池，而不是总是终止未来需要的实例容量。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 的温水池。	2022 年 2 月 24 日
指南更改 (p. 355)	Amazon EC2 Auto Scaling 控制台已通过其他选项进行了更新，以帮助您在启用跳过匹配并指定所需配置的情况下开启实例刷新。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 开启或取消实例刷新（控制台）。	2022 年 2 月 3 日

预测性扩展策略的自定义指标 (p. 355)	您现在可以选择是否在创建预测性扩展策略时使用自定义指标。您还可以使用指标数学来进一步自定义策略中包含的指标。有关更多信息，请参阅 使用自定义指标进行高级预测性扩展配置。	2021 年 11 月 24 日
新的按需分配策略 (p. 355)	现在，创建使用混合实例策略的 Auto Scaling 组时，您可以选择是否根据价格（最低价格的实例类型优先）启动按需型实例。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的分配策略。	2021 年 10 月 27 日
基于属性的实例类型选择 (p. 355)	Amazon EC2 Auto Scaling 新增支持基于属性的实例类型选择。您可以将实例要求表达为一组属性，例如 vCPU、内存和存储，而不是手动选择实例类型。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的使用基于属性的实例类型选择创建 Auto Scaling 组。	2021 年 10 月 27 日
支持按标签筛选组 (p. 355)	现在，当您使用 `describe-auto-scaling-groups` 命令检索有关 Auto Scaling 组的信息时，可以使用标签筛选条件筛选 Auto Scaling 组。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的使用标签筛选 Auto Scaling 组。	2021 年 10 月 14 日
指南更改 (p. 355)	Amazon EC2 Auto Scaling 控制台已更新，以帮助您使用 Amazon Lambda 创建自定义终止策略。控制台文档已相应地进行了修订。有关更多信息，请参阅使用不同的终止策略（控制台）。	2021 年 10 月 14 日
支持将启动配置复制到启动模板 (p. 355)	现在，您可以将 Amazon 区域中的所有启动配置从 Amazon EC2 Auto Scaling 控制台复制到新启动模板。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的将启动配置复制到启动模板。	2021 年 8 月 9 日
扩展实例刷新功能 (p. 355)	现在，您可以在通过将所需的配置添加到 `start-instance-refresh` 命令来替换实例时包含更新（例如新版本的启动模板）。您也可以通过启用跳过匹配来跳过替换已包含所需配置的实例。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的基于实例刷新替换 Auto Scaling 实例。	2021 年 8 月 5 日

支持自定义终止策略 (p. 355)	现在，您可以使用 Amazon Lambda 创建自定义终止策略。有关更多信息，请参阅使用 Lambda 创建自定义终止策略。用于指定终止策略的文档进行了相应的更新。	2021 年 7 月 29 日
指南更改 (p. 355)	Amazon EC2 Auto Scaling 控制台已更新并通过其他功能进行增强以帮助您在指定时区内创建计划操作。适用于计划扩展的文档已进行相应的修订。	2021 年 6 月 3 日
启动配置中的 gp3 卷 (p. 355)	您现在可以在块储存设备映射中为启动配置指定 gp3 卷。	2021 年 6 月 2 日
支持预测性扩展 (p. 355)	现在，您可以使用预测性扩展来使用扩展策略主动扩展您的 Amazon EC2 Auto Scaling 组。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 的预测性扩展。通过此更新，AutoScalingServiceRolePolicy 托管式策略现在包含调用 `cloudwatch:GetMetricData` API 操作所需的权限。	2021 年 5 月 19 日
指南更改 (p. 355)	您现在可以从 GitHub 访问生命周期钩子的示例模板。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 生命周期挂钩。	2021 年 4 月 9 日
支持温水池 (p. 355)	现在，您可以通过向 Auto Scaling 组添加温水池来平衡首次启动时间较长的应用程序的性能（最大限度地减少冷启动）和成本（停止过度配置实例容量）。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 的温水池。	2021 年 4 月 8 日
支持检查点 (p. 355)	现在，您可以将检查点添加到实例刷新中，以分阶段替换实例并在特定时间点对您的实例执行验证。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的将检查点添加到实例刷新。	2021 年 3 月 18 日

指南更改 (p. 355)	改进了将 EventBridge 与 Amazon EC2 Auto Scaling 事件和生命周期钩子结合使用的文档。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的将 Amazon EC2 Auto Scaling 与 EventBridge 结合使用和教程：配置调用 Lambda 函数的生命周期钩子。	2021 年 3 月 18 日
支持本地时区 (p. 355)	现在，您可以在本地时区中通过将 `--time-zone` 选项添加到 put-scheduled-update-group-action 命令来创建重复计划操作。如果您的时区遵守夏令时 (DST)，则重复操作会自动调整夏令时。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的计划扩展。	2021 年 3 月 9 日
扩展混合实例策略的功能 (p. 355)	现在，当您使用混合实例策略时，您可以为竞价型容量设置实例类型的优先级。Amazon EC2 Auto Scaling 会尽最大努力尝试满足优先级，但首先会针对容量进行优化。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的具有多个实例类型和购买选项的 Auto Scaling 组。	2021 年 3 月 8 日
已删除组的扩展活动 (p. 355)	现在，您可以通过将 `--include-deleted-groups` 选项添加到 describe-scaling-activities 命令来查看已删除 Auto Scaling 组的扩展活动。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 故障排除。	2021 年 2 月 23 日
控制台改进 (p. 355)	您现在可以从 Amazon EC2 Auto Scaling 控制台创建和附加 Application Load Balancer 或 Network Load Balancer。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的创建和附加新的 Application Load Balancer 或 Network Load Balancer（控制台）。	2020 年 11 月 24 日
多个网络接口 (p. 355)	现在，您可以为指定多个网络接口的 Auto Scaling 组配置启动模板。有关更多信息，请参阅 VPC 中的网络接口。	2020 年 11 月 23 日
多个启动模板 (p. 355)	多个启动模板现在可以与 Auto Scaling 组一起使用。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南中的为实例类型指定不同的启动模板。	2020 年 11 月 19 日

Gateway Load Balancer (p. 355)	更新的指南显示了如何将网关负载均衡器附加到 Auto Scaling 组，以确保 Amazon EC2 Auto Scaling 启动的设备实例从负载均衡器自动注册并取消注册。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Elastic Load Balancing 类型和将负载均衡器附加到 Auto Scaling 组。	2020 年 11 月 10 日
最大实例生命周期 (p. 355)	现在，您可以将最大实例生命周期减少到一天（86,400 秒）。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的基于最大实例生命周期替换 Auto Scaling 实例。	2020 年 11 月 9 日
容量再平衡 (p. 355)	您可以在 Amazon EC2 发出再平衡建议时将您的 Auto Scaling 组配置为启动替换竞价型实例。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 容量再平衡。	2020 年 11 月 4 日
实例元数据服务版本 2 (p. 355)	使用启动配置时，您可能需要使用实例元数据服务版本 2，这是一种面向会话的方法，用于请求实例元数据。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的配置实例元数据选项。	2020 年 7 月 28 日
指南更改 (p. 355)	以下各种改进和新控制台步骤：控制在扩展过程中终止哪些 Auto Scaling 实例、监控您的 Auto Scaling 实例和组、Amazon EC2 Auto Scaling 用户指南的启动模板和启动配置部分。	2020 年 7 月 28 日
实例刷新 (p. 355)	在更改配置时启动实例刷新，可以更新 Auto Scaling 组中的所有实例。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的基于实例刷新替换 Auto Scaling 实例。	2020 年 6 月 16 日
指南更改 (p. 355)	以下各种改进：根据最大实例生命周期替换 Auto Scaling 实例、包括多个实例类型和购买选项的 Auto Scaling 组、根据 Amazon SQS 进行扩展和 Amazon EC2 Auto Scaling 用户指南的标记 Auto Scaling 组和实例部分。	2020 年 5 月 6 日

指南更改 (p. 355)	IAM 文档的各种改进。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的启动模板支持和 Amazon EC2 Auto Scaling 基于身份的策略示例。	2020 年 3 月 4 日
禁用扩缩策略 (p. 355)	您现在可以禁用和重新启用扩展策略。使用此功能，您可以在临时禁用扩展策略的同时保留配置详细信息，这样以后便可以再次启用该策略。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的禁用 Auto Scaling 组的扩展策略。	2020 年 2 月 18 日
增加了通知功能 (p. 355)	现在，当您的 Auto Scaling 组由于缺少安全组或启动模板而无法向外扩展时，Amazon EC2 Auto Scaling 会将事件发送到您的 Amazon Health Dashboard。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon Health Dashboard Amazon EC2 Auto Scaling 通知。	2020 年 2 月 12 日
指南更改 (p. 355)	以下各种改进和更正：Amazon EC2 Auto Scaling 如何与 IAM 结合使用、Amazon EC2 Auto Scaling 基于身份的策略示例、与加密卷结合使用时所需的 CMK 密钥策略以及 Amazon EC2 Auto Scaling 用户指南的监控您的 Auto Scaling 实例和组部分。	2020 年 2 月 10 日
指南更改 (p. 355)	改进了使用实例权重的 Auto Scaling 组的文档。了解如何在使用“容量单位”衡量所需容量时使用扩展策略。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的扩展策略的工作原理和扩展调整类型。	2020 年 2 月 6 日
新增“安全性”章节 (p. 355)	Amazon EC2 Auto Scaling 用户指南中新的安全章节可帮助您了解如何在使用 Amazon EC2 Auto Scaling 时应用责任共担模式。作为此更新的一部分，已将用户指南的“控制对您的 Amazon EC2 Auto Scaling 资源的访问权限”一章替换为更实用的新部分，即适用于 Amazon EC2 Auto Scaling 的 Identity and Access Management。	2020 年 2 月 4 日

实例类型建议 (p. 355)	Amazon Compute Optimizer 提供了 Amazon EC2 实例建议，以帮助您提高性能和/或节省资金。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的获取实例类型建议。	2019 年 12 月 3 日
专用主机和主机资源组 (p. 355)	已更新演示如何创建指定主机资源组的启动模板的指南。这样，您可以使用启动模板创建 Auto Scaling 组，该模板指定要在专用主机上使用的 BYOL AMI。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的为 Auto Scaling 组创建启动模板。	2019 年 12 月 3 日
对 Amazon VPC 终端节点的支持 (p. 355)	您现在可以在 VPC 和 Amazon EC2 Auto Scaling 之间建立私有连接。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 Amazon EC2 Auto Scaling 和接口 VPC 终端节点。	2019 年 11 月 22 日
最大实例生命周期 (p. 355)	现在，您可以通过指定实例可以使用的最长时间来自动替换实例。如果任何实例接近此限制，Amazon EC2 Auto Scaling 将逐步替换它们。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的基于最大实例生命周期替换 Auto Scaling 实例。	2019 年 11 月 19 日
实例权重 (p. 355)	对于具有多个实例类型的 Auto Scaling 组，您现在可以选择指定每个实例类型对组容量贡献的容量单位数。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的 Amazon EC2 Auto Scaling 的实例权重。	2019 年 11 月 19 日
最小实例类型数量 (p. 355)	您不再需要为竞价型实例、按需型实例和预留实例组指定其他实例类型。对于所有 Auto Scaling 组，最小值现在为一个实例类型。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的具有多个实例类型和购买选项的 Auto Scaling 组。	2019 年 9 月 16 日
支持新的竞价型分配策略 (p. 355)	Amazon EC2 Auto Scaling 现在支持新的“容量优化”竞价型分配策略，该策略使用根据可用竞价型容量选择的最优竞价型实例池满足您的请求。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的具有多个实例类型和购买选项的 Auto Scaling 组。	2019 年 8 月 12 日

指南更改 (p. 355)	改进了服务相关角色和与加密卷结合使用时所需的 CMK 密钥策略主题中的 Amazon EC2 Auto Scaling 文档。	2019 年 8 月 1 日
支持标记增强功能 (p. 355)	Amazon EC2 Auto Scaling 现在将标签添加到 Amazon EC2 实例作为启动实例的相同 API 调用的一部分。有关更多信息，请参阅标记 Auto Scaling 组和实例。	2019 年 7 月 26 日
指南更改 (p. 355)	改进了暂停和恢复扩展流程主题中的 Amazon EC2 Auto Scaling 文档。更新了客户托管策略示例以包括一个示例策略，该策略允许用户仅将特定的后缀自定义服务相关角色传递到 Amazon EC2 Auto Scaling。	2019 年 6 月 13 日
支持新的 Amazon EBS 功能 (p. 355)	在启动模板主题中增加了对新的 Amazon EBS 功能的支持。在从快照还原时更改 EBS 卷的加密状态。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的为 Auto Scaling 组创建启动模板。	2019 年 5 月 13 日
指南更改 (p. 355)	改进了以下各节中的 Amazon EC2 Auto Scaling 文档：控制在扩展过程中终止哪些 Auto Scaling 实例、Auto Scaling 组、包括多个实例类型和购买选项的 Auto Scaling 组以及 Amazon EC2 Auto Scaling 的动态扩展。	2019 年 3 月 12 日
支持组合实例类型和购买选项 (p. 355)	在单个 Auto Scaling 组中通过多个购买选项（竞价型、按需型和预留实例）和实例类型预置和自动扩展实例。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的具有多个实例类型和购买选项的 Auto Scaling 组。	2018 年 11 月 13 日
更新了有关基于 Amazon SQS 扩展的主题 (p. 355)	更新了指南，介绍如何使用自定义指标，根据 Amazon SQS 队列中的需求变化来扩展 Auto Scaling 组。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的基于 Amazon SQS 的扩展。	2018 年 7 月 26 日

下表介绍了在 2018 年 7 月以前对 Amazon EC2 Auto Scaling 文档的一些重要更改。

功能	描述	发行日期
对目标跟踪扩展策略的支持	只需几个步骤就可为应用程序设置动态扩展。有关更多信息，请参阅 Amazon EC2 Auto Scaling 的目标跟踪扩展策略。	2017 年 7 月 12 日

功能	描述	发行日期
支持资源级权限	创建 IAM 策略以控制资源级访问。有关更多信息，请参阅控制对 Amazon EC2 Auto Scaling 资源的访问。	2017 年 5 月 15 日
监控改进	Auto Scaling 组指标不再需要您启用详细监控。您现在可以启用组指标集合和从控制台中的 Monitoring (监控) 选项卡上查看指标图形。有关更多信息，请参阅使用 Amazon CloudWatch 监控 Auto Scaling 组和实例。	2016 年 8 月 18 日
支持 Application Load Balancer	将一个或多个目标组附加到新的或现有的 Auto Scaling 组。有关更多信息，请参阅将负载均衡器附加到 Auto Scaling 组。	2016 年 8 月 11 日
生命周期挂钩的事件	Amazon EC2 Auto Scaling 在 EventBridge 调用生命周期钩子时向 EventBridge 发送事件。有关更多信息，请参阅在 Auto Scaling 组扩展时获取 EventBridge。	2016 年 2 月 24 日
实例保护	阻止 Amazon EC2 Auto Scaling 在缩减时选择终止特定实例。有关更多信息，请参阅实例保护。	2015 年 12 月 07 日
分步扩展策略	创建扩展策略，使您能根据警报的严重程度来扩展。有关更多信息，请参阅扩展策略类型。	2015 年 7 月 6 日
更新负载均衡器	将负载均衡器附加到现有 Auto Scaling 组或从中分离。有关更多信息，请参阅将负载均衡器附加到 Auto Scaling 组。	2015 年 6 月 11 日
支持 ClassicLink	将 Auto Scaling 组中的 EC2-Classic 实例连接到 VPC，让这些链接的 EC2-Classic 实例与 VPC 中的实例能够利用私有 IP 地址进行通信。有关更多信息，请参阅将 EC2-Classic 实例链接到 VPC。	2015 年 1 月 19 日
生命周期钩子	在您对新启动的实例或正在终止的实例执行操作时，请将这些实例保持在等待状态。有关更多信息，请参阅 Amazon EC2 Auto Scaling 生命周期钩子。	2014 年 7 月 30 日
分离实例	从 Auto Scaling 组分离实例。有关更多信息，请参阅从 Auto Scaling 组分离 EC2 实例。	2014 年 7 月 30 日
将实例置于 Standby 状态	将处于 `InService` 状态的实例置于 `Standby` 状态。有关更多信息，请参阅从 Auto Scaling 组临时删除实例。	2014 年 7 月 30 日
管理标签	使用 Amazon Web Services Management Console 管理您的 Auto Scaling 组。有关更多信息，请参阅标记 Auto Scaling 组和实例。	2014 年 5 月 01 日
支持专用实例	通过在创建启动配置时指定部署租期属性启动专用实例。有关更多信息，请参阅实例部署租期。	2014 年 4 月 23 日
从 EC2 实例创建组或启动配置	使用 EC2 实例创建 Auto Scaling 组或启动配置。有关使用 EC2 实例创建启动配置的信息，请参阅使用 EC2 实例创建启动配置。有关使用 EC2 实例创建 Auto Scaling 组的信息，请参阅使用 EC2 实例创建 Auto Scaling 组。	2014 年 1 月 02 日
附加实例	您可以通过将 EC2 实例附加到现有 Auto Scaling 组，为该实例启用自动扩展。有关更多信息，请参阅将 EC2 实例附加到 Auto Scaling 组。	2014 年 1 月 02 日
查看账户限制	查看对您的账户的 Auto Scaling 资源限制。有关更多信息，请参阅 Auto Scaling 限制。	2014 年 1 月 02 日

功能	描述	发行日期
Amazon EC2 Auto Scaling 的控制台支持	使用 Amazon Web Services Management Console 访问 Amazon EC2 Auto Scaling。有关更多信息，请参阅 Amazon EC2 Auto Scaling 入门。	2013 年 12 月 10 日
分配公有 IP 地址	将公有 IP 地址分配给启动至 VPC 的实例。有关更多信息，请参阅在 VPC 中启动 Auto Scaling 实例。	2013 年 9 月 19 日
实例终止策略	为 Amazon EC2 Auto Scaling 指定实例终止策略，用于终止 EC2 实例。有关更多信息，请参阅控制在扩展期间终止哪些 Auto Scaling 实例。	2012 年 9 月 17 日
对 IAM 角色的支持	用 IAM 实例配置文件启动 EC2 实例。您可以使用此功能为实例指定 IAM 角色，允许您的应用程序安全地访问其他 Amazon Web Services 服务。有关更多信息，请参阅使用 IAM 角色启动 Auto Scaling 实例。	2012 年 6 月 11 日
支持竞价型实例	使用启动配置启动竞价型实例。有关更多信息，请参阅为容错和灵活的应用程序请求竞价型实例。	2012 年 6 月 7 日
对组和实例进行标记	标记 Auto Scaling 组并指定将标签也应用于在创建标签之后启动的 EC2 实例。有关更多信息，请参阅标记 Auto Scaling 组和实例。	2012 年 1 月 26 日
Amazon SNS 支持	每当 Amazon EC2 Auto Scaling 启动或终止 EC2 实例时，都使用 Amazon SNS 接收通知。有关更多信息，请参阅在 Auto Scaling 组扩展时获取 SNS 通知。 Amazon EC2 Auto Scaling 还增加了以下新功能： • 使用 cron 语法设置重复扩展活动的能力。有关更多信息，请参阅 `PutScheduledUpdateGroupAction` API 操作。 • 增加了新的配置设置，使您无需将已启动的实例添加到负载均衡器 (LoadBalancer) 即可进行扩展。有关更多信息，请参阅 `ProcessType` API 数据类型。 • `DeleteAutoScalingGroup` 操作中的 `ForceDelete` 标志指示 Amazon EC2 Auto Scaling 删除 Auto Scaling 组及其关联实例，而无需先等待实例终止。有关更多信息，请参阅 `DeleteAutoScalingGroup` API 操作。	2011 年 7 月 20 日
计划扩展操作	增加了对计划扩展操作的支持。有关更多信息，请参阅 Amazon EC2 Auto Scaling 的计划扩展。	2010 年 12 月 2 日
Amazon VPC 支持	添加了对 Amazon VPC 的支持。有关更多信息，请参阅在 VPC 中启动 Auto Scaling 实例。	2010 年 12 月 2 日
支持 HPC 集群	增加了对于高性能计算 (HPC) 集群的支持。	2010 年 12 月 2 日
支持运行状况检查	增加了对使用 Amazon EC2 Auto Scaling 托管式 EC2 实例进行 Elastic Load Balancing 运行状况检查的支持。有关更多信息，请参阅结合使用 ELB 运行状况检查和 Auto Scaling。	2010 年 12 月 2 日
支持 CloudWatch 警报	删除了较旧的触发机制，并重新设计了 Amazon EC2 Auto Scaling 以便使用 CloudWatch 警报功能。有关更多信息，请参阅 Amazon EC2 Auto Scaling 的动态扩展。	2010 年 12 月 2 日
暂停和恢复扩缩	增加了对暂停和恢复扩展过程的支持。	2010 年 12 月 2 日

功能	描述	发行日期
对 IAM 的支持	增加了对 IAM 的支持。有关更多信息，请参阅控制对 Amazon EC2 Auto Scaling 资源的访问。	2010 年 12 月 2 日