



FleetIQ 开发人员指南

# Amazon GameLift



版本

# Amazon GameLift: FleetIQ 开发人员指南

Copyright © 2024 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、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他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。

# Table of Contents

什么是 Amazon GameLift FleetIQ ? .....	1
FleetIQ 的工作原理 .....	1
亚马逊 GameLift FleetIQ 逻辑 .....	2
关键资源和组件 .....	5
游戏架构 .....	6
补充本地托管 .....	6
生命周期和状态 .....	8
竞价平衡过程 .....	8
游戏服务器组的生命周期 .....	10
游戏服务器的生命周期 .....	12
最佳实操 .....	14
Amazon GameLift FleetIQ 特征 .....	16
Amazon GameLift FleetIQ 的定价 .....	17
设置 .....	18
支持的软件 .....	18
设置您的 AWS 账户 .....	19
创建一个 AWS 账户 .....	19
管理 Amazon GameLift FleetIQ 的用户权限 .....	21
为跨服务交互创建 IAM 角色 .....	25
为 FleetIQ 准备游戏 .....	31
集成步骤 .....	31
管理游戏服务器组 .....	33
创建游戏服务器组 .....	33
更新游戏服务器组 .....	34
跟踪游戏服务器组实例 .....	34
集成游戏服务器 .....	34
注册游戏服务器 .....	34
更新游戏服务器状态 .....	35
注销游戏服务器 .....	36
集成游戏客户端 .....	36
让 Amazon GameLift FleetIQ 选择一台游戏服务器 .....	36
选择您自己的游戏服务器 .....	36
使用 CloudWatch 监控 GPU .....	38
Amazon GameLift FleetIQ 参考 .....	40

---

服务 API 参考 ( AWS 软件开发工具包 ) .....	40
Amazon GameLift FleetIQ API 操作 .....	40
可用编程语言 .....	41
使用 FleetIQ 实现高安全性 .....	42
发行说明和软件开发工具包版本 .....	43
Amazon GameLift 的所有指南 .....	44
AWS 术语表 .....	45
.....	xlvi

# 什么是 Amazon GameLift FleetIQ ？

Amazon GameLift FleetIQ 优化了对 Amazon Elastic Compute Cloud (Amazon EC2) 竞价型实例的使用，用于基于云的游戏托管。借助 Amazon GameLift FleetIQ，您可以直接使用 Amazon EC2 和 Amazon EC2 Auto Scaling 中的托管资源，同时利用 Amazon GameLift 的优化为玩家提供便宜、弹性的游戏托管。Amazon EC2 竞价型实例虽然折扣很大，但通常不适合游戏托管，因为可用性会波动，而且有可能出现[中断](#)。Amazon GameLift FleetIQ 大大缓解了这些限制，使得使用低成本竞价型实例可以用于游戏托管。

在使用 Amazon GameLift 管理游戏托管时，也可以使用 FleetIQ 优化。有关 Amazon GameLift 托管选项的信息，请参阅 [Amazon GameLift 开发人员指南](#)。

Amazon GameLift FleetIQ 游戏托管解决方案专为符合以下条件的游戏开发人员而设计：

- 已有 AWS 部署或想直接使用 Amazon EC2，而不是通过完全托管的 Amazon GameLift 服务使用。Amazon GameLift FleetIQ 可与您在 AWS 账户中管理的 EC2 自动扩缩组配合使用，使您可以完全访问自己的 EC2 实例和组。您还可以与其他 AWS 服务集成，包括 Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS) 和 AWS Shield Advanced。
- 拥有现有的本地游戏托管，并希望将容量扩展到云。借助 Amazon GameLift FleetIQ，您可以构建一个混合部署系统，该系统使用您的本地容量，并根据需要逐步增加 AWS 云容量。

## 准备好开始操作 Amazon GameLift FleetIQ 了吗？

- 参加 AWS Skill Builder 上的[在游戏服务器上使用 Amazon GameLift FleetIQ](#)课程，了解如何使用 Amazon GameLift FleetIQ 进行游戏。有关相关课程的概述，请参阅[游戏技术学习计划](#)。有些课程有不同的语言版本。
- 按照中的说明进行操作[Amazon GameLift FleetIQ 集成步骤](#)

## 亚马逊 GameLift FleetIQ 是如何运作的

Amazon GameLift FleetIQ 解决方案是一个游戏托管层，它补充了你通过 Amazon EC2 和 Auto Scaling 获得的全套计算资源管理工具。除了提供一系列特定于游戏托管的功能外，Amazon GameLift FleetIQ 还提供了额外的逻辑层，使使用低成本竞价型实例进行游戏托管成为可能。该解决方案允许您直接管理您的 Amazon EC2 和 Auto Scaling 资源，并根据需要与其他 AWS 服务集成。

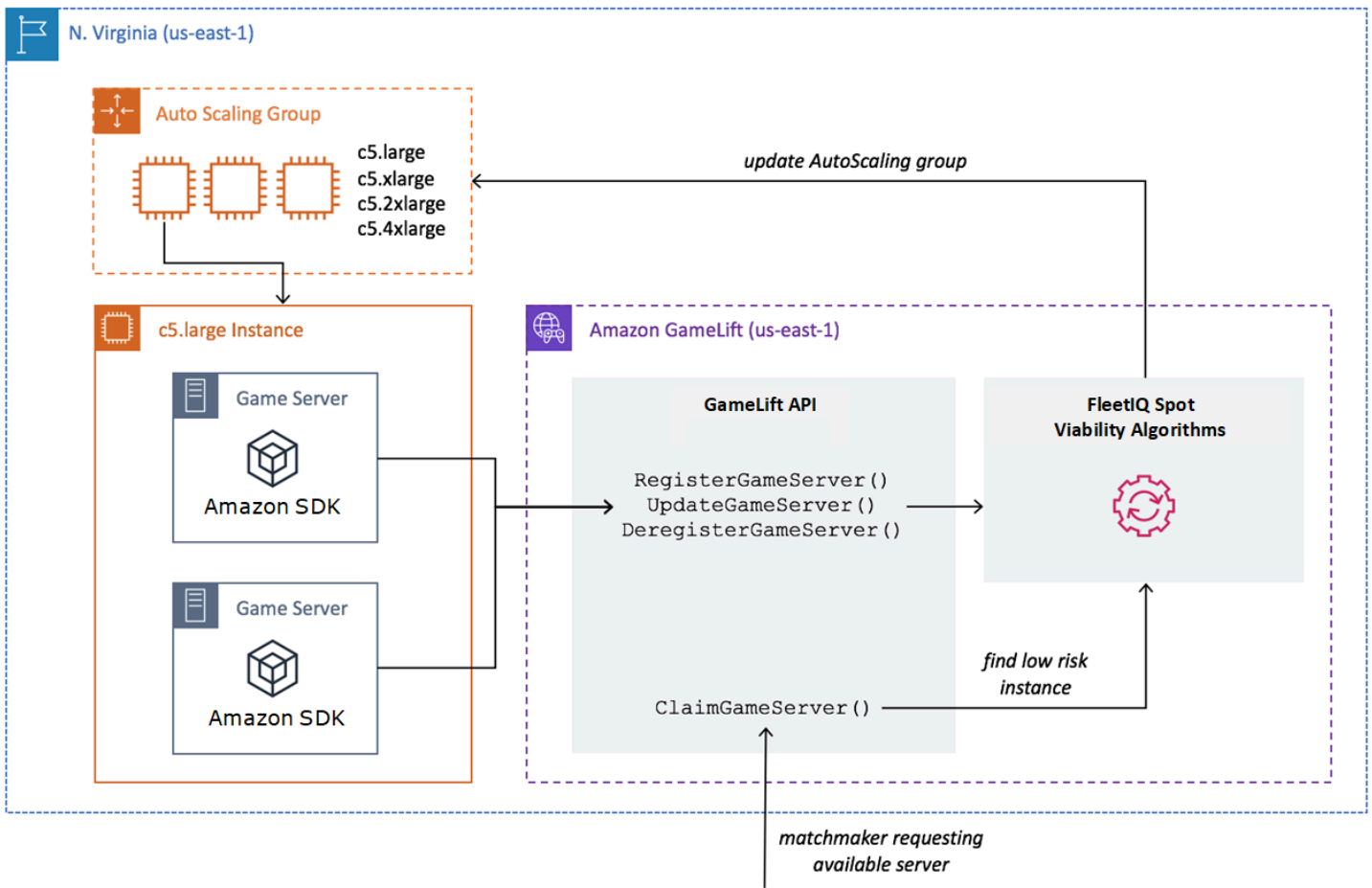
使用 Amazon GameLift FleetiQ 时，您可以像往常一样准备启动 Amazon EC2 实例：使用游戏服务器软件制作亚马逊系统映像 (AMI)，创建亚马逊 EC2 启动模板，并为 Auto Scaling 组定义配置设置。但是，您无需直接创建 Auto Scaling 组，而是使用您的 Amazon GameLift EC2 和 Auto Scaling 资源和配置创建 Amazon FleetiQ 游戏服务器组。此操作会提示 Amazon GameLift FleetiQ 同时创建游戏服务器组和相应的 Auto Scaling 组。游戏服务器组与自动扩缩组关联并管理自动扩缩组的某些方面。

创建自动扩缩组后，您将具有 Amazon EC2 和 Auto Scaling 资源的完全访问权限。您可以更改 Auto Scaling 组的配置，添加多级扩展策略或负载均衡器，以及与其他 AWS 服务集成。您可以直接连接到组中的实例。作为其优化逻辑的一部分，Amazon GameLift FleetiQ 还会定期更新某些 Auto Scaling 组属性。您可以跟踪自动扩缩组部署的所有实例的可用性状态。

您可以随时暂停游戏服务器组的 Amazon GameLift FleetiQ 活动。您还可以选择删除游戏服务器组，但保留对应的自动扩缩组。

## 亚马逊 GameLift FleetiQ 逻辑

下图说明了亚马逊 GameLift FleetiQ 在与亚马逊 EC2 合作托管游戏时所扮演的角色。其主要目标是找到最好的游戏服务器来托管游戏会话，并为玩家提供最佳的游戏体验。Amazon GameLift FleetiQ 将最佳资源定义为那些以最低成本提供最高游戏托管可行性的资源。Amazon GameLift FleetiQ 通过两种关键方式实现这一目标：首先，在 Auto Scaling 组中仅允许可行的实例类型，其次是在该组的可用资源中有效地放置新的游戏会话。



## 使用最佳实例类型填充自动扩缩组

自动扩缩组的任务是启动新实例并停用旧实例，同时维护托管资源集合并扩展该集合以满足玩家需求。为此，自动扩缩组依赖于您所需的实例类型列表。Amazon GameLift FleetiQ 的工作是不断检查这些所需实例类型的可行性，并更新 Auto Scaling 组的列表。此过程称为实例平衡。它可确保自动扩缩组中的实例不断刷新，以便始终仅使用当前可行的实例类型。

Amazon GameLift FleetiQ 通过以下方式影响 Auto Scaling 组选择最佳实例类型的方式：

- 它决定竞价型和/或按需型实例的使用情况。Amazon GameLift FleetiQ 游戏服务器组配置了平衡策略，这会影响 Auto Scaling 组使用竞价型和/或按需实例的方式。由于可用性波动和潜在的**中断**，竞价型实例的成本较低，Amazon FleetiQ 将游戏服务器托管的限制降至 GameLift 最低。按需型实例成本更高昂，但在您需要时可提供更可靠的可用性。
- 它限制新实例只能在可行的实例类型上启动。Amazon GameLift FleetiQ 游戏服务器组维护您所需实例类型的主列表。实例平衡过程使用预测算法，持续评估列表中每种所需的实例类型，以确定游戏托管的可行性，该算法会查看该实例类型的最新可用性和中断率。根据此次评估，Amazon GameLift FleetiQ 会不断更新 Auto Scaling 组的所需实例类型列表，以仅包括当前可行的实例类型。

- 它会标记不可行的实例类型的现有实例。Amazon GameLift FleetiQ 会识别 Auto Scaling 组中当前不可行的实例类型的现有实例。这些实例标记为耗尽，这意味着将终止这些实例并使用新实例替换。对于已启用游戏服务器保护的实例，终止将推迟到任何活动的游戏会话正常结束。

随着自动扩缩组启动和停用实例，它将持续维护针对游戏托管进行优化的集合，即使低成本竞价型实例类型的可用性不断波动也是如此。平衡活动仅在具有活动实例的游戏服务器组上进行。在[竞价平衡过程](#)中了解有关此过程如何工作的详细信息。

## 有效地放置游戏会话

Amazon GameLift FleetiQ 会跟踪游戏服务器组中所有活跃的游戏服务器，并使用这些信息来确定新游戏会话和玩家的最佳位置。

要让 Amazon GameLift FleetiQ 能够跟踪游戏服务器，您的游戏服务器软件必须报告其状态。您的自定义 AMI 控制在每个实例上启动和停止新游戏服务器进程的方式。当新的游戏服务器启动时，它会向 Amazon GameLift FleetiQ 注册，这表明它已准备好托管游戏会话。在注册后，游戏服务器定期报告其运行状况以及当前是否托管游戏会话。当游戏服务器关闭时，它会在亚马逊 Fle GameLift etiQ 上注销注册。

要开始新的游戏会话，您的游戏客户端（或媒人或其他客户端服务）会向 Amazon GameLift FleetiQ 发送游戏服务器请求。Amazon GameLift FleetiQ 会找到可用的游戏服务器，为新的游戏会话申领该服务器，然后使用游戏服务器 ID 和连接信息进行响应。然后，游戏会提示游戏服务器更新其状态，并为进入的玩家启动新的游戏会话。

在选择游戏服务器来托管新游戏会话时，Amazon GameLift FleetiQ 使用以下决策过程通过可行的低成本竞价型实例来优化布局：

1. 在可能的情况下，Amazon GameLift FleetiQ 会在已经托管其他游戏会话的实例上放置新的游戏会话。通过打包（但不过载）某些实例并保持其他实例为空闲状态，自动扩缩组能够在不需要空闲实例时快速缩减空闲实例，从而降低托管成本。
2. Amazon GameLift FleetiQ 会忽略那些被标记为耗尽资源（即不适用于游戏托管）的实例。这些实例仅为持现有的游戏会话而保持运行。除非没有其他游戏服务器可用，否则它们不能用于新的游戏会话。
3. Amazon GameLift FleetiQ 可识别在可行实例上运行的所有可用游戏服务器。

您可以为游戏服务器组启用游戏会话保护，以防止自动扩缩组终止具有正在运行游戏会话的实例。



## 关键资源和组件

在使用 Amazon GameLift FleetiQ 设置游戏托管资源之前，请在您的 AWS 账户中创建以下资源。作为最佳做法，在通过游戏服务器组使用这些资源之前，使用这些资源开发和测试游戏服务器部署。

- **亚马逊机器映像 (AMI)**。AMI 是您希望随 Amazon EC2 实例一起启动的特定软件配置的模板。对于游戏托管，AMI 包括操作系统、游戏服务器二进制文件或容器，以及游戏服务器所需的其他运行时软件。有关创建 AMI 的更多信息，请参阅 [Amazon EC2 用户指南中的亚马逊系统映像](#)。AMI 是特定于区域的。您可以将 AMI 从一个区域复制到另一个区域，如同在《Amazon EC2 用户指南》中的 [复制 AMI](#) 所述。
- **Amazon EC2 启动模板**。启动模板提供了在自动扩缩组中启动和管理实例的说明。它指定 AMI，提供合适的实例类型列表，设置网络、安全性和其他属性。有关创建启动模板的更多信息，请参阅《Amazon EC2 用户指南》中的 [从启动模板启动实例](#)。启动模板特定于区域。
- **AWS IAM 角色**。IAM 角色定义了一组权限，允许对 AWS 资源进行有限访问。可信实体（例如其他 AWS 服务）可以担任该角色并继承其权限。使用 Amazon GameLift FleetiQ 时，您必须为一个 IAM 角色提供托管策略，该策略允许 Amazon GameLift FleetiQ 在您的账户中创建和访问 Auto Scaling 组和 EC2 实例资源。AWS IAM 角色不是特定于区域的。

Amazon GameLift FleetiQ 直接管理以下资源，并直接管理这些资源。

- **GameLift 游戏服务器组**。游戏服务器组包含配置设置，这些设置定义了 Amazon GameLift FleetiQ 如何与相应的 Auto Scaling 组合作以提供低成本的游戏托管。游戏服务器组是区域特定的。当您在某个区域中创建游戏服务器组时，系统会在同一区域的 AWS 账户中自动创建一个新的 Auto Scaling 组。游戏服务器组与 Auto Scaling 组关联，并且有权（通过担任 IAM 角色）管理和修改其部分设置。游戏服务器组是一个长寿命的资源；开发人员应预期很少创建它们。游戏服务器组也是托管在 Auto Scaling 组中的实例上并在 Amazon GameLift FleetiQ 上注册的游戏服务器的功能分组资源。
- **GameLift 游戏服务器**。游戏服务器资源表示在与 Amazon GameLift FleetiQ 游戏服务器组关联的实例上运行的游戏执行。该资源是在游戏服务器向 Amazon GameLift FleetiQ 注册并识别其所属的游戏服务器组时创建的。Amazon GameLift FleetiQ 会跟踪每台已注册游戏服务器的使用状态和申领状态，从而使其能够监控游戏服务器的可用性。游戏服务器特定于区域，因为它们与特定于区域的游戏服务器组相关联。当您的游戏请求新的游戏服务器时，它会指定游戏服务器组和区域。

这些资源是通过 Amazon GameLift FleetiQ 资源创建的。它们在您的 AWS 账户中创建，您对它们具有完全控制权限。

- Amazon EC2 自动扩缩组。Auto Scaling 组启动和管理 EC2 实例集合，并自动扩展组容量。在 Amazon GameLift FleetiQ 中，游戏服务器组和 one-to-one Auto Scaling 组之间存在关系。虽然您可以更新 Auto Scaling 组的所有设置，但 Amazon GameLift FleetiQ 会定期覆盖和更新某些设置，这是其逻辑的一部分，以平衡竞价型实例的游戏托管可行性。有关更多信息，请参阅 [AutoScalingGroup](#) Amazon EC2 Auto Scaling 用户指南。Auto Scaling 组是特定于区域的；它们创建在与游戏服务器组相同的区域中。
- Amazon EC2 实例。实例是云中的虚拟服务器。实例类型具有指定计算、内存、磁盘和网络资源的特定硬件配置。它们通常由具有 AMI 的自动扩缩组启动。实例可以是 Spot 或按需实例，具体取决于可用性。使用 Amazon GameLift FleetiQ，实例可以运行一个或多个游戏服务器进程，每个进程都可以托管多个游戏会话。实例特定于区域，因为它们与特定于区域的自动扩缩组相关联。

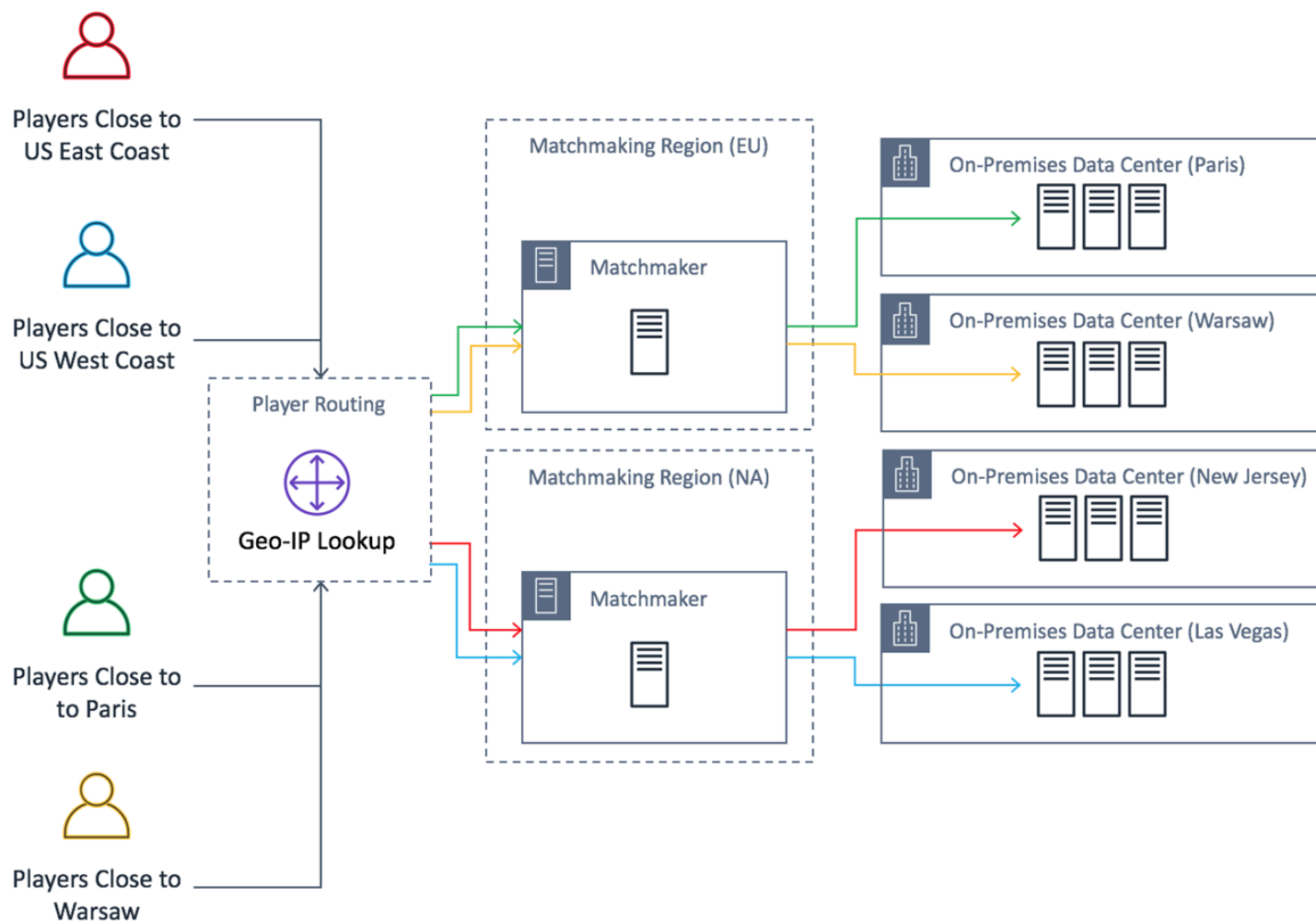
## Amazon GameLift FleetiQ 游戏架构

### 补充本地托管

Amazon GameLift FleetiQ 旨在重复使用您现有的游戏后端，包括您可能已经具备的任何玩家 geo-IP 路由、对战或游戏大厅服务。以下示例说明 Amazon GameLift FleetiQ 如何适合现有的本地部署。

#### Example

在此示例中，游戏托管最初由四个专有数据中心处理，以便在北美和欧洲托管玩家。根据它们的大致实际位置，玩家会被路由到两个区域匹配器中的一个。匹配器会按技能和延迟对玩家进行分组，然后将他们放到附近的游戏服务器上，以最大限度地减少滞后。

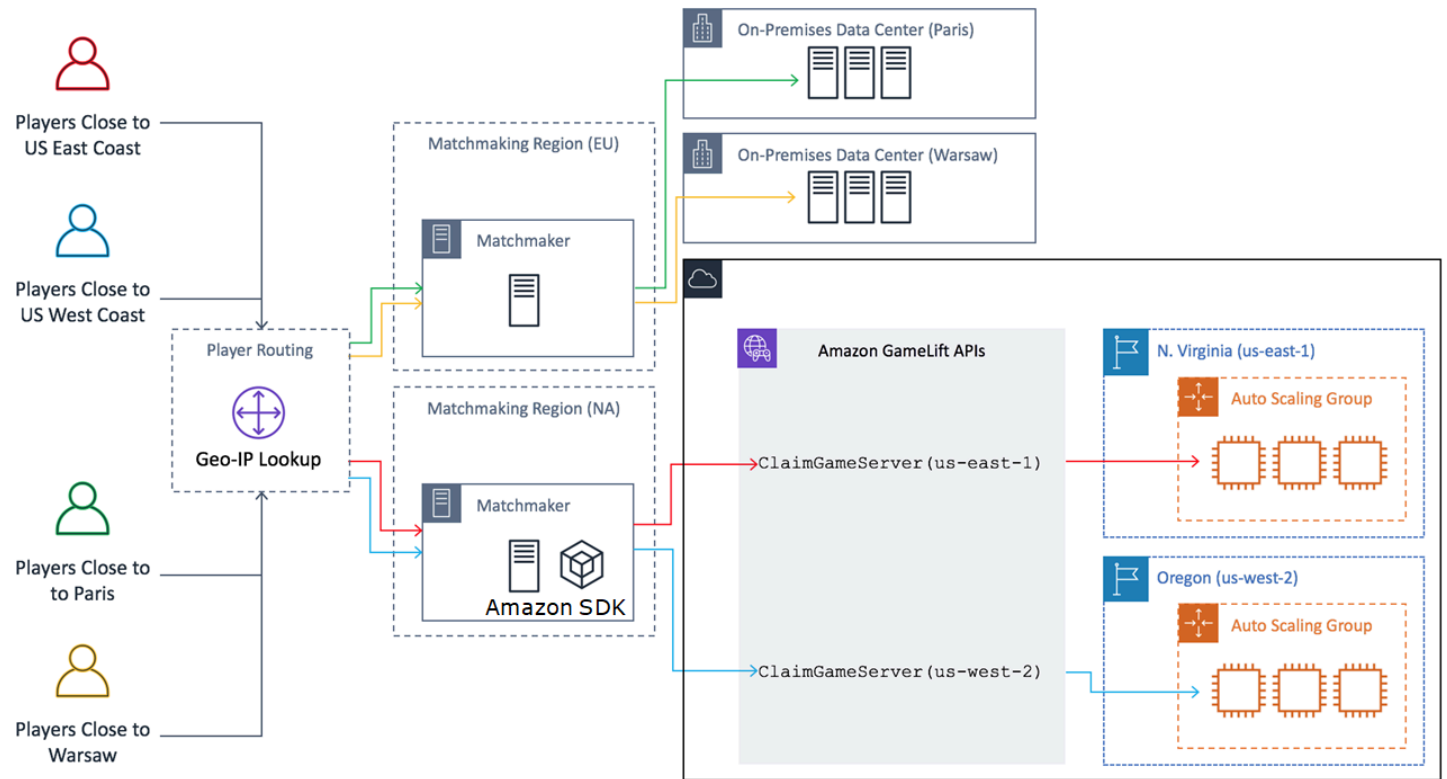


游戏开发人员希望用 Amazon GameLift FleetIQ 提供的服务器替换他们的北美游戏服务器。首先，他们会对游戏服务器进行次要更新以便将其与 Amazon GameLift FleetIQ 一起使用，然后创建亚马逊机器映像 (AMI)。此映像将安装在为游戏部署的每个 EC2 实例上。该映像包含游戏服务器、依赖关系以及为玩家运行游戏会话所需的任何其他内容。

AMI 准备就绪后，开发人员创建了两个 Amazon GameLift FleetIQ 游戏服务器组，每个 AWS 北美地区一个 ( us-east-1 和 us-west-2 )。开发人员传入启动模板 ( 提供 AMI )、所需实例类型列表以及该组的其他配置设置。所需实例类型的列表告知 Amazon GameLift FleetIQ，在搜索适用于游戏托管的竞价型实例时要使用哪些类型。

最后，开发人员将 AWS 软件开发工具包和 Amazon GameLift FleetIQ 集成到他们的北美对战构建器中，当一群新的玩家需要服务器容量来进行游戏会话时，它会调用 Amazon GameLift FleetIQ。Amazon GameLift FleetIQ 查找带有可用游戏服务器的竞价型实例，将其保留给玩家，并提供服务器连接信息。玩家连接到服务器，玩游戏，然后断开连接。要开始一个新的游戏，玩家重新进入对战，这会提示 Amazon GameLift FleetIQ 查找另一个可用的游戏服务器。每个新的游戏请求都会触发

Amazon GameLift FleetIQ 搜索和选择中断几率低的游戏服务器。因此，即使竞价型实例的可用性随着时间推移而波动，Amazon GameLift FleetIQ 也会不断将玩家从不可用于游戏托管的游戏服务器重定向到它处。



## Amazon GameLift FleetIQ 的生命周期和状态

### 竞价平衡过程

Amazon GameLift FleetIQ 会定期平衡具有竞价型实例的自动扩缩组中的实例。对于使用 ON\_DEMAND\_ONLY 平衡策略或没有任何活动实例的游戏服务器组，此过程不处于活动状态。

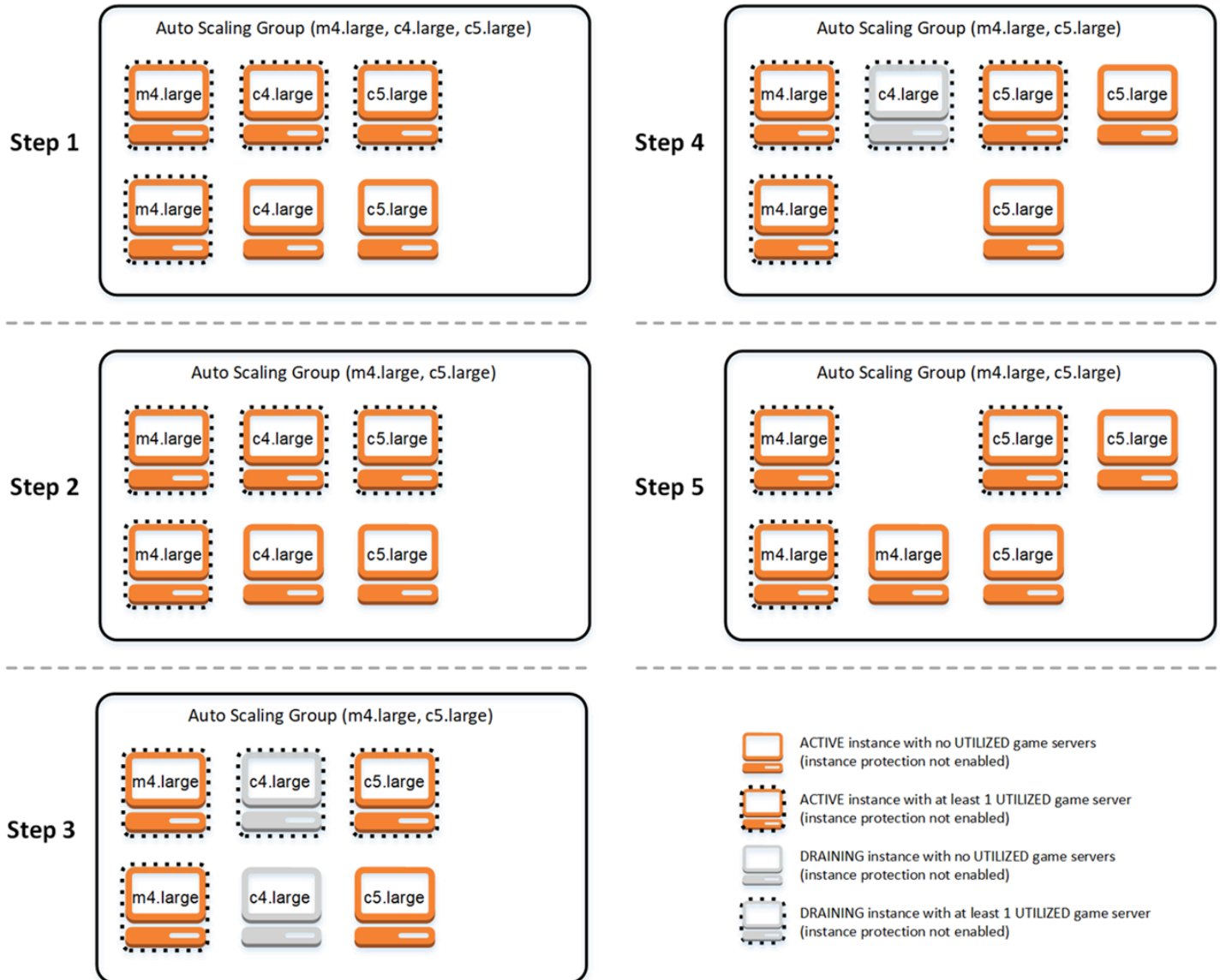
竞价平衡有两个关键目标：

- 仅使用对游戏托管具有高可行性的竞价型实例类型持续刷新组。
- 使用多种可行的实例类型（在可能时），以减少游戏服务器意外中断的影响

Amazon GameLift FleetIQ 通过评估组的实例类型并删除更可能导致游戏服务器中断的实例来平衡。为了避免在平衡过程中终止具有活动游戏的实例，最好是为生产中的游戏服务器组启用游戏服务器保护。

## Example

以下示例说明自动扩缩组中的实例如何受到竞价平衡的影响。



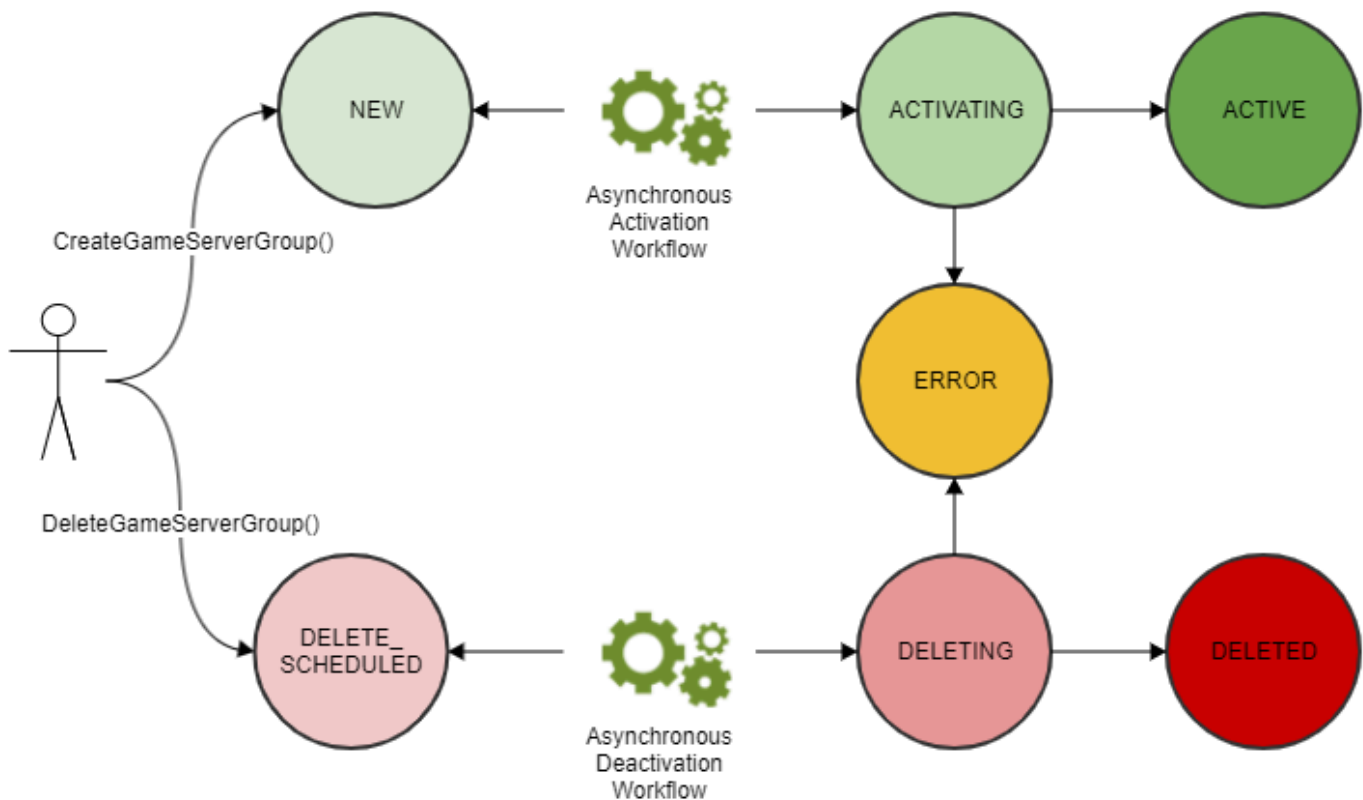
- 步骤 1. 通过游戏服务器组，关联的自动扩缩组设置为在启用游戏服务器保护的情况下启动类型为 m4.large、c4.large 和 c5.large 的实例。自动扩缩组已启动一个平衡集合，其中包含每种类型的两个竞价型实例。四个实例至少有一个处于 UTILIZED（被利用）状态的游戏服务器（显示为虚线边框），而两个实例当前不支持玩游戏。
- 步骤 2. Amazon GameLift FleetIQ 评估所有三种实例类型的当前游戏托管可行性。评估确定 c4.large 实例类型存在不可接受的游戏服务器中断的可能性。Amazon GameLift FleetIQ 会立即更新自动扩缩组配置，暂时将 c4.large 从实例类型列表中删除，从而阻止启动其他 c4.large 实例。

- 步骤 3 Amazon GameLift FleetIQ 确定类型 c4.large 的现有实例，并采取措施将它们从组中删除。作为第一步，在 c4.large 实例上运行的所有游戏服务器都标记为耗尽。只有在没有其他游戏服务器可用的情况下，才能作为最后手段认领耗尽实例上的游戏服务器。此外，将触发具有耗尽实例的自动扩缩组以启动新实例来替换它们。
- 步骤 4. 随着新的可行实例上线，自动扩缩组将终止耗尽的实例。这种替换可确保该组的所需容量得到保持。要终止的第一个实例是没有已利用的游戏服务器且关闭了游戏服务器保护的 c4.large 实例。它被替换为一个新的 c5.large 实例。
- 步骤 5. 对于具有游戏服务器保护的耗尽实例而言，在其游戏服务器支持玩游戏时，这些实例将继续运行。游戏结束后，一旦新的 m4.large 实例启动以取代剩余的 c5.large 实例的位置，后者就会终止。

由于此过程，自动扩缩组将保持其所需容量，同时此组将从使用三种实例类型平衡为使用两种。Amazon GameLift FleetIQ 继续评估最初的实例类型列表，以确定游戏托管的可行性。当 c4.large 再次被视为可行的实例类型时，将更新自动扩缩组以包含全部三种实例类型。该组随着时间推移自然地平衡。

## 游戏服务器组的生命周期

游戏服务器组经历以下生命周期，包括预配置和状态更新。游戏服务器组预计是一个长寿命的资源。



- 您可以通过调用 Amazon GameLift API `CreateGameServerGroup()` 并传入 EC2 启动模板和配置设置来创建游戏服务器组。为响应呼叫，将创建一个新的游戏服务器组并将其置于 **NEW**（新）状态。
- Amazon GameLift FleetIQ 激活异步激活工作流，将游戏服务器组状态转换为 **ACTIVATING**。该工作流程启动对基础资源的创建，包括 Amazon EC2 Auto Scaling 组和具有提供的 AMI 的 EC2 实例。
- 如果预配置因任何原因失败，游戏服务器组将被置于 **ERROR**（错误）状态。要获取其他错误信息以帮助调试故障原因，请在处于错误状态的游戏服务器组上调用 `DescribeGameServerGroup()`。
- 如果预配置成功，则游戏服务器组将转换为 **ACTIVE**（活动）状态。此时，将通过注册到 Amazon GameLift FleetIQ 的游戏服务器启动实例。定期评估该组的实例类型以确定游戏托管的可行性，并根据需要进行平衡。Amazon GameLift FleetIQ 还会跟踪组中活跃游戏服务器的状态，并响应对游戏服务器的请求。
- 可以通过使用组标识符调用 `DeleteGameServerGroup()` 来删除游戏服务器组。此操作将游戏服务器组置于 **DELETE\_SCHEDULED** 状态。只能计划删除处于 **ACTIVE**（活动）或 **ERROR**（错误）状态的游戏服务器组。

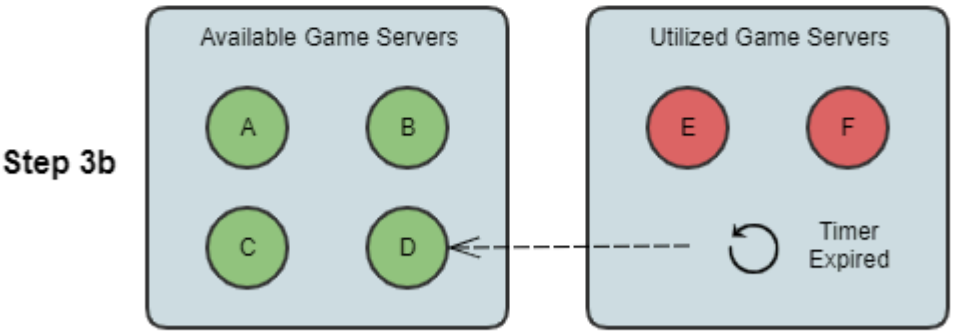
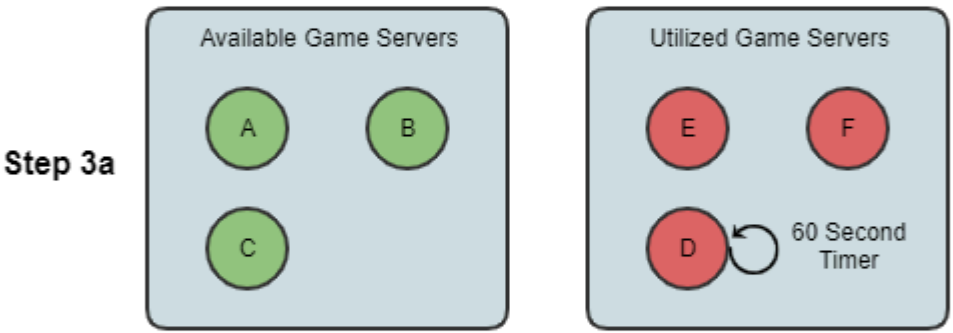
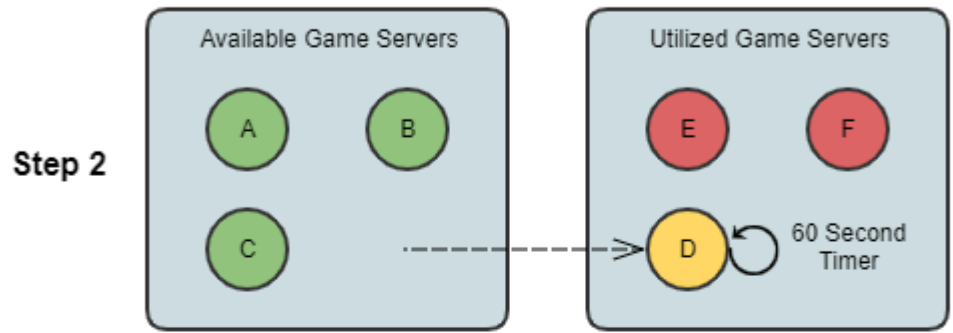
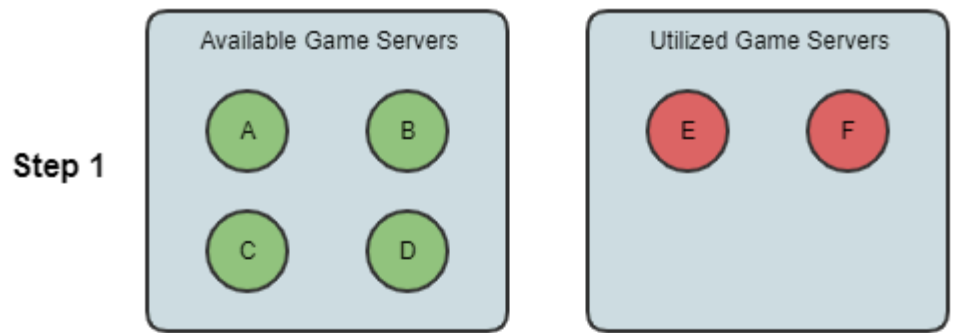
- Amazon GameLift FleetIQ 激活异步停用工作流程以响应 DELETE\_SCHEDULED 状态，从而将游戏服务器组状态转换为 DELETING。您可以选择仅删除游戏服务器组，或者同时删除游戏服务器组和关联的自动扩缩组。
- 如果停用因任何原因失败，游戏服务器组将进入 ERROR ( 错误 ) 状态。要获取其他错误信息以帮助调试故障原因，请在处于错误状态的游戏服务器组上调用 DescribeGameServerGroup()。
- 如果停用成功，游戏服务器组将转换为 DELETED ( 已删除 ) 状态。


## 游戏服务器的生命周期

使用 Amazon GameLift FleetIQ，游戏服务器经历以下生命周期，包括预配置和状态更新。游戏服务器预计是短寿命的资源。最佳实操是，游戏服务器应在游戏会话结束后注销，而不是重新用于其他游戏会话。这种方法有助于确保可用的游戏服务器始终在可用于游戏托管的成本最低的资源上运行。

- 当在 Amazon GameLift FleetIQ 关联的自动扩缩组中的某个实例上运行的游戏服务器进程调用 Amazon GameLift API RegisterGameServer() 以通知 Amazon GameLift FleetIQ 它已准备好托管玩家和玩游戏时，将创建游戏服务器资源。游戏服务器具有两种状态来跟踪其当前可用性：
  - 利用状态跟踪游戏服务器当前是否支持玩游戏。此状态最初设置为 AVAILABLE ( 可用 )，表示它已准备好接受新的游戏。一旦游戏服务器被游戏占用，此状态就会设置为 UTILIZED ( 已利用 )。
  - 认领状态可跟踪游戏服务器是否被认领用于即将开始的游戏。处于 CLAIMED ( 已认领 ) 状态的游戏服务器表示它已被游戏客户端 ( 或游戏服务，如匹配器 ) 暂时保留。此状态可防止 Amazon GameLift FleetIQ 向多个请求者提供相同的游戏服务器。认领状态为空的游戏服务器可供认领。
- 下图说明游戏服务器的利用状态和认领状态在其生命周期内如何变化。





-  Utilization Status is AVAILABLE, no Claim Status
-  Utilization Status is AVAILABLE, Claim Status is CLAIMED
-  Utilization Status is UTILIZED, Claim Status can be either

- 步骤 1. 一个游戏服务器组有六个已注册的游戏服务器。其中四个的利用状态为 AVAILABLE ( A、B、C 和 D ) ，两个目前处于 UTILIZED 状态 ( E 和 F ) 。
- 步骤 2. 游戏客户端或对战系统调用 Amazon GameLift API ClaimGameServer() 来请求新的游戏服务器。此请求会提示 Amazon GameLift FleetIQ 搜索可用的游戏服务器 (D) ，并将其申领状态设置为 CLAIMED ，持续 60 秒。Amazon GameLift FleetIQ 在回应其请求时提供了游戏服务器的连接信息 ( IP 地址和端口 ) 以及其他可选的游戏特定数据。由于尚未在游戏服务器上开始玩游戏，因此它的利用状态仍然保持为 AVAILABLE ，但无法通过其他请求来认领。
- 步骤 3a. 使用提供的连接信息，游戏客户端可以连接到游戏服务器并启动游戏。必须在 60 秒内触发游戏服务器 (D) ，才能通过调用 Amazon GameLift API UpdateGameServer() 将其利用状态更改为 UTILIZED。
- 步骤 3b. 如果游戏服务器的利用状态在 60 秒后未更新，认领计时器将过期，认领状态将重置为空白。游戏服务器 (D) 将返回到可用和未认领的游戏服务器池中。
- 游戏服务器上的游戏完成并且玩家断开连接后，游戏服务器资源将被移除。在关闭之前，游戏服务器进程调用 Amazon GameLift API DeregisterGameServer() ，以通知 Amazon GameLift FleetIQ 其离开游戏服务器组的游戏服务器池。

## Amazon GameLift FleetIQ 最佳实操

Amazon GameLift FleetIQ 是一个低级逻辑层，可帮助您管理用于游戏托管的 Amazon EC2 资源。特别是，Amazon GameLift FleetIQ 通过最大限度地减少游戏会话中断的可能性，优化了适用于游戏托管的竞价型实例的使用。它还提供基本的游戏托管功能，以跟踪可用的游戏服务器，并将游戏路由到低成本、高可行性的游戏服务器。

Amazon GameLift FleetIQ 作为独立特征，不提供与完整 Amazon GameLift 托管解决方案一起提供的高级特征，后者也使用 FleetIQ 来尽可能降低托管成本。如果您需要对战、基于延迟的玩家路由、游戏会话和玩家会话管理以及版本控制等功能，请查看 Amazon GameLift 解决方案。

以下是一些可以帮助您从 Amazon GameLift FleetIQ 中获得最大利益的最佳实操。

- 将 Amazon GameLift FleetIQ 用于基于会话的游戏。当 Amazon GameLift FleetIQ 不断地将玩家引导到最不可能出现游戏会话中断的实例时，其效果最佳。保持长寿命的会话会干扰 Amazon GameLift FleetIQ 平衡过程，从而增加游戏会话可能被中断的几率。理想的工作流程是让玩家从对战 ( 或服务器选择 ) 到进入游戏。游戏结束后，玩家会返回到对战，并路由到新实例上的另一个游戏服务器。我们建议将 Amazon GameLift FleetIQ 用于会话时间少于两小时的游戏。
- 提供多种实例类型以供选择。设置游戏服务器组时，您提供要使用的实例类型列表。您包含的实例类型越多，Amazon GameLift FleetIQ 使用高可行的竞价型实例进行游戏托管的灵活性就越大。例如，

您可以列出同一个实例系列中的多种大小 ( c5.large、c5.xlarge、c5.2xlarge、c5.4xlarge )。对于大型实例，您可以在每个实例上运行更多的游戏服务器，从而可能降低成本。对于小型实例，自动缩放可以更快地响应玩家需求的变化。请记住，所需实例类型列表未按优先顺序排列，自动扩缩组将使用可行实例类型的平衡来保持该组的弹性。

- 在所有实例类型上测试您的游戏。确保游戏服务器在为游戏服务器组配置的每个实例类型上正常运行。
- 使用实例容量加权。如果您将游戏服务器组配置为使用某个实例大小范围 ( 如 c5.2xlarge、c5.4xlarge、c5.12xlarge )，请包括每种实例类型的容量加权信息。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](#)。
- 使用 Amazon GameLift FleetIQ 进行游戏会话。将几组玩家放置在游戏服务器上时，请使用 Amazon GameLift API `ClaimGameServer()`。Amazon GameLift FleetIQ 避免将玩家置于游戏会话中断几率更高的实例上。
- 向 Amazon GameLift FleetIQ 报告游戏服务器状态。使用 Amazon GameLift API `UpdateGameServer()` 定期报告服务器运行状况和利用状态。保持准确的游戏服务器状态有助于 Amazon GameLift FleetIQ 更高效地放置游戏。它还有助于避免在竞价平衡活动期间终止具有活动游戏的实例。
- 设置自动扩展策略。您可以创建目标跟踪扩展策略，以根据玩家利用情况和预期需求维护您的主机容量。Amazon GameLift FleetIQ 指标 `PercentUtilizedGameServers` 用于衡量当前正在使用的托管容量。大多数游戏希望维护未使用的游戏服务器的缓冲区，以便新玩家能够快速进入游戏。您可以创建一个保持特定缓冲区大小的扩展策略，随着玩家需求的波动添加或删除实例。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的 [目标跟踪扩展策略](#)。
- 在开发和生产环境中使用不同的 AWS 账户。跨账户将您的开发配置与生产配置分开，可以减少影响现场玩家的配置错误风险。
- 为使用中的游戏服务器组启用游戏会话保护。为了保护玩家，请打开游戏会话保护，防止活动游戏会话因扩展或平衡活动而提前终止。
- 在与 Amazon GameLift FleetIQ 集成之前，在 EC2 上测试游戏。我们建议首先您在 EC2 上启动并运行您的游戏来进行调整。然后，您可以使用相同的启动模板和 AMI 创建游戏服务器组。

如果您使用的是 Kubernetes，我们建议您首先将标准 EC2 实例添加到 Kubernetes 集群中，然后使用您在 Kubernetes 集群中为工作线程节点创建的启动模板来创建游戏服务器组。如果您使用的是 EKS，请单独创建 EKS 集群和游戏服务器组。对于游戏服务器组，请将 EKS 优化的 AMI 与合适的用户数据以及用于 EKS 集成的启动模板配置结合使用。在 [Amazon EKS 优化 Linux AMI](#) 指南中可查看有关 EKS 工作线程节点和 EKS 优化 AMI 的更多信息。

- 使用游戏服务器组平衡策略 **ON\_DEMAND\_ONLY** 获得可靠的游戏服务器可用性。此平衡策略生效后，将不使用竞价型实例。这是一个有用的工具，可以确保服务器在您最需要的时候（例如在特征发布或其他特殊活动期间）的可用性。您可以根据需要将游戏服务器组从竞价型策略切换到按需型策略。

此外，请查看以下 AWS 最佳实操：

- [Amazon EC2 的最佳实操](#)
- [Amazon EC2 的最佳实操](#)

## Amazon GameLift FleetIQ 特征

- 优化了竞价平衡。Amazon GameLift FleetIQ 定期评估您的实例类型，并替换由于游戏会话中断可能性较高而视为不可用的竞价型实例。当您的 EC2 自动扩缩组停用旧实例并启动新实例时，该组将持续刷新当前可用于游戏托管的实例类型。
- 最佳玩家路线。Amazon GameLift FleetIQ API 将新游戏会话引导到最不可能中断的竞价型实例。此外，游戏会话也被打包到较少的实例上，这改进了 EC2 自动扩缩组缩减不必要的资源并降低成本的能力。
- 根据玩家使用情况自动扩展。Amazon GameLift FleetIQ 将游戏服务器利用率数据作为 Amazon CloudWatch 指标发布。您可以使用这些指标来自动扩展您的可用托管资源，以跟踪实际玩家需求并降低托管成本。
- 直接管理 Amazon EC2 实例。保持对您 AWS 账户中的 EC2 实例和 EC2 自动扩缩组的完全控制。这意味着您可以设置实例启动模板、维护 EC2 自动扩缩组配置以及与其他 AWS 服务集成。作为竞价平衡活动的一部分，Amazon GameLift FleetIQ 会定期更新一些 EC2 自动扩缩组属性。您可以根据需要暂时覆盖这些设置或暂停 Amazon GameLift FleetIQ 活动。
- 支持多种游戏服务器可执行文件格式。Amazon GameLift FleetIQ 支持目前在 Amazon EC2 上运行的所有格式，包括 Windows、Linux、容器和 Kubernetes。有关支持的操作系统和运行时系统的列表，请参阅 [Amazon EC2 常见问题解答](#)。
- 多种类型的托管资源。借助 Amazon GameLift FleetIQ，您可以访问用于托管游戏服务器的大量实例类型。（可用性因 AWS 区域而异。）这意味着，您可以将您的游戏服务器与适当的 CPU、内存、存储和网络容量组合进行配对，以便为您的玩家提供最佳的游戏体验。
- 覆盖全球。Amazon GameLift FleetIQ 已在 15 个区域上市，包括中国。这种覆盖范围意味着玩家无论身处何处都能够以最小的游戏延迟访问您提供的游戏服务器。有关区域的完整列表，请参阅《AWS 一般参考》中的 [Amazon GameLift 端点和限额](#)。

# Amazon GameLift FleetIQ 的定价

Amazon GameLift 按使用时长对实例收费，按传输的数据量收取带宽费用。有关 Amazon GameLift 的费用和价格的完整列表，请参阅 [Amazon GameLift 定价](#)。

有关计算托管游戏或通过 Amazon GameLift 对战的成本的信息，请参阅 [生成 Amazon GameLift 定价估算值](#)，其中描述了如何使用 [AWS Pricing Calculator](#)。

# Amazon GameLift FleetiQ 设置

本节中的主题有助于设置任务，包括如何设置 AWS 账户以使用 Amazon GameLift FleetiQ 服务。

## 主题

- [亚马逊 GameLift FleetiQ 支持的软件](#)
- [为亚马逊 GameLift FleetiQ 设置 AWS 账户](#)

## 亚马逊 GameLift FleetiQ 支持的软件

Amazon GameLift FleetiQ 用于部署 64 位多人游戏服务器、客户端和游戏服务，以便在亚马逊上托管。EC2 此解决方案支持以下环境：

### 游戏服务器的操作系统

您可以将 Amazon GameLift FleetiQ 与在支持的任何操作系统上运行的游戏服务器一起使用。EC2 这包括亚马逊 Linux、Ubuntu、Windows 服务器、红帽企业 Linux、Linux 企业服务器、Fedora、Debian、CentOS、Gentoo SUSE Linux、Oracle Linux、Oracle Linux 和 Free。BSD 在 [Amazon EC2 功能中查看当前 EC2 功能和支持](#)。

### 使用容器

如果你的游戏服务器使用容器，亚马逊 GameLift FleetiQ 支持与 Kubernetes、亚马逊弹性容器服务（亚马逊）和亚马逊 Elastic Kubernetes Service ECS 集成。EKS 请在 [AWS 上的容器](#) 中查看更多信息。

### 游戏开发环境

游戏客户端和服务端需要一些集成才能与 Amazon GameLift FleetiQ 服务进行通信。游戏 API 拨打 AWS SDK。 [下载 AWS SDK 或查看 Amazon GameLift API 参考文档](#)。

支持 AWS SDK Amazon GameLift 的，有以下语言版本。有关开发环境支持的信息，请参阅每种语言的文档。

- C++ ( [SDK 文档](#) ) ( [亚马逊 GameLift](#) )
- Java ( [SDK 文档](#) ) ( [亚马逊 GameLift](#) )
- 。 NET ( [SDK 文档](#) ) ( [亚马逊 GameLift](#) )

- Go ([SDKdocs](#)) ([亚马逊 GameLift](#))
- Python ([SDK文档](#)) ([亚马逊 GameLift](#))
- Ruby ([SDK文档](#)) ([亚马逊 GameLift](#))
- PHP ([SDK文档](#)) ([亚马逊 GameLift](#))
- JavaScript/Node.js ([SDK文档](#)) ([亚马逊 GameLift](#))

## 为亚马逊 GameLift FleetiQ 设置 AWS 账户

要将 Amazon GameLift FleetiQ 与 Amazon EC2、Auto Scaling AWS 和其他服务配合使用，您必须设置 AWS 账户 具有所需访问权限的。完成以下任务：

- 如果您还没有可在 Amazon GameLift FleetiQ 上使用的 AWS 账户，请创建一个新账户。请参阅 [创建一个 AWS 账户](#)。
- 为用户和用户组设置 GameLift 特定于 Amazon FleetiQ 的权限。请参阅 [管理 Amazon GameLift FleetiQ 的用户权限](#)。
- 创建 IAM 角色以允许亚马逊 GameLift 和您的 Amazon EC2 资源进行交互。请参阅 [为跨服务交互创建 IAM 角色](#)。

## 创建一个 AWS 账户

创建并设置一个 AWS 账户 用于亚马逊 GameLift FleetiQ 的。创建 AWS 账户不收取任何费用。

### 主题

- [注册获取 AWS 账户](#)
- [创建具有管理访问权限的用户](#)

## 注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

### 要注册 AWS 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建 AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行 [需要根用户访问权限的任务](#)。

AWS 注册过程完成后会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 <https://aws.amazon.com/> 并选择我的账户来查看当前的账户活动并管理您的账户。

## 创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

### 保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。 [AWS Management Console](#) 在下一页上，输入您的密码。

要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的 [以根用户身份登录](#)。

2. 为您的根用户启用多重身份验证 (MFA)。

有关说明，请参阅 [IAM 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备 \(控制台\)](#)。

### 创建具有管理访问权限的用户

1. 启用 IAM Identity Center

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 [启用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，为用户授予管理访问权限。

有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅 [《用户指南》IAM Identity Center 目录中的使用默认设置配置 AWS IAM Identity Center 用户访问权限](#)。

### 以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。



有关使用 IAM Identity Center 用户[登录的帮助](#)，请参阅[AWS 登录 用户指南中的登录 AWS 访问门户](#)。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[创建权限集](#)。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[添加组](#)。

## 管理 Amazon GameLift FleetiQ 的用户权限

根据需要创建更多用户或将 Amazon GameLift FleetiQ 访问权限扩展到现有用户。使用 Amazon GameLift FleetiQ 游戏服务器组以及相关的 Amazon EC2 和 Auto Scaling 服务的用户必须拥有访问这些服务的权限。

作为最佳实操（[IAM 中的安全最佳实操](#)），请为所有用户应用最低权限权限。您可以为各个用户或用户组设置权限，并按服务、操作或资源限制用户访问权限。

根据您的管理 AWS 账户中用户的方式，按照以下说明设置用户权限。如果您使用 IAM 用户，则作为最佳实操，请始终向角色或用户组授予权限，而不是向个人用户授予权限。

- [用户的权限语法](#)
- [与一起使用的其他权限语法 AWS CloudFormation](#)

要提供访问权限，请为您的用户、组或角色添加权限：

- 中的用户和群组 AWS IAM Identity Center：

创建权限集合。按照《AWS IAM Identity Center 用户指南》中[创建权限集](#)的说明进行操作。

- 通过身份提供商在 IAM 中托管的用户：

创建适用于身份联合验证的角色。按照《IAM 用户指南》中[为第三方身份提供商创建角色（联合身份验证）](#)的说明进行操作。

- IAM 用户：

- 创建您的用户可以担任的角色。按照《IAM 用户指南》中[为 IAM 用户创建角色](#)的说明进行操作。
- (不推荐使用) 将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中[向用户添加权限 \(控制台\)](#)中的说明进行操作。

## 亚马逊 GameLift Fleetiq\_Policy

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Action":
      [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition":
      {
        "StringEquals":
        {
          "iam:PassedToService": "gamelift.amazonaws.com"
        }
      }
    },
    {
      "Action":
      [
        "iam:CreateServiceLinkedRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:*:iam::*:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
    },
    {
      "Action":
      [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:CreateOrUpdateTags",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:ExitStandby",
        "autoscaling:PutLifecycleHook",
```

```
        "autoscaling:PutScalingPolicy",
        "autoscaling:ResumeProcesses",
        "autoscaling:SetInstanceProtection",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling>DeleteAutoScalingGroup"
    ],
    "Effect": "Allow",
    "Resource": "*"
},
{
    "Action":
    [
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeSubnets",
        "ec2:RunInstances",
        "ec2:CreateTags"
    ],
    "Effect": "Allow",
    "Resource": "*"
},
{
    "Action":
    [
        "events:PutRule",
        "events:PutTargets"
    ],
    "Effect": "Allow",
    "Resource": "*"
}
]
```

## 的额外权限 AWS CloudFormation

如果您使用 AWS CloudFormation 管理游戏托管资源，请将 AWS CloudFormation 权限添加到策略语法中。

```
{
    "Action": [
        "autoscaling:DescribeLifecycleHooks",
        "autoscaling:DescribeNotificationConfigurations",
        "ec2:DescribeLaunchTemplateVersions"
    ]
}
```

```
"Effect": "Allow",
"Resource": "*"
}
```

## 为用户设置编程式访问权限

如果用户想在 AWS 外部进行交互，则需要编程访问权限 AWS Management Console。授予编程访问权限的方式取决于正在访问的用户类型 AWS。

要向用户授予编程式访问权限，请选择以下选项之一。

哪个用户需要编程式访问权限？	目的	方式
人力身份 (在 IAM Identity Center 中管理的用户)	使用临时证书签署向 AWS CLI、AWS 软件开发工具包或 AWS API 发出的编程请求。	按照您希望使用的界面的说明进行操作。 <ul style="list-style-type: none"> <li>有关的 AWS CLI，请参阅 <a href="#">《AWS Command Line Interface 用户指南》AWS IAM Identity Center 中的配置 AWS CLI 以使用</a>。</li> <li>有关 AWS 软件开发工具包、工具和 AWS API，请参阅 <a href="#">《软件开发工具包和 AWS 工具参考指南》中的 IAM 身份中心身份验证</a>。</li> </ul>
IAM	使用临时证书签署向 AWS CLI、AWS 软件开发工具包或 AWS API 发出的编程请求。	按照 IAM 用户指南中的 <a href="#">将临时证书与 AWS 资源配合使用</a> 中的说明进行操作。
IAM	(不推荐使用) 使用长期凭证签署向 AWS CLI、AWS 软件开发工具包或 AWS API 发出的编程请求。	按照您希望使用的界面的说明进行操作。 <ul style="list-style-type: none"> <li>有关信息 AWS CLI，请参阅用户指南中的 <a href="#">使用 IAM 用户证书进行身份验证</a>。AWS Command Line Interface</li> </ul>

哪个用户需要编程式访问权限？	目的	方式
		<ul style="list-style-type: none"><li>• 有关 AWS SDK 和工具，请参阅 <a href="#">S AWS DK 和工具参考指南中的使用长期凭证进行身份验证</a>。</li><li>• 有关 AWS API，请参阅 <a href="#">IAM 用户指南中的管理 IAM 用户的访问密钥</a>。</li></ul>

如果您使用访问密钥，请参阅[管理 AWS 访问密钥的最佳实践](#)。

## 为跨服务交互创建 IAM 角色

为了让 Amazon GameLift FleetiQ 与您的 Amazon EC2 实例和 Auto Scaling 组配合使用，您必须允许这些服务相互交互。这是通过在您的 AWS 账户中创建 IAM 角色并分配一组有限权限来完成的。每个角色还指定能够担任此角色的服务。

设置以下角色：

- [为 Amazon GameLift FleetiQ 创建角色](#)更新您的 Amazon EC2 资源。
- [为 Amazon EC2 创建角色](#)用于与亚马逊 GameLift FleetiQ 沟通的资源。

## 为 Amazon GameLift FleetiQ 创建角色

此角色允许 Amazon GameLift FleetiQ 访问和修改您的 Amazon EC2 实例、Auto Scaling 组和生命周期挂钩，这是其竞价平衡和自动扩展活动的一部分。

使用 IAM 控制台或 AWS CLI 为 Amazon GameLift FleetiQ 创建角色并附加具有必要权限的托管策略。有关 IAM 角色和托管策略的更多信息，请参阅[为 AWS 服务创建角色](#)和[AWS 托管策略](#)。

### Console

这些步骤描述了如何 GameLift 使用为 Amazon 创建具有托管策略的服务角色 AWS Management Console。

1. 登录 [IAM 控制台](#)，然后选择角色: 创建角色。

2. 对于 Select type of trusted entity ( 选择受信任实体的类型 ) , 选择 AWS service ( 服务 ) 。
3. 对于选择用例, 请 GameLift 从服务列表中进行选择。在“选择您的用例”下, 将自动选择相应的 Amazon GameLift 用例。要继续, 请选择下一步: 权限。
4. 列表附加的权限策略应包含一个策略: GameLiftGameServerGroupPolicy。如果未显示此策略, 请检查筛选条件或使用搜索功能将其添加到角色中。您可以查看策略的语法 ( 选择 ▶ 图标以展开 ) , 但不能更改语法。创建角色后, 您可以更新角色并附加其他策略以添加或删除权限。

对于设置权限边界, 保留默认设置 ( 创建没有权限边界的角色 ) 。这是一个不必要的高级设置。要继续, 请选择下一步: 标签。

5. 添加标签是用于资源管理的可选设置。例如, 您可能希望向此角色添加标签, 以便按角色跟踪特定于项目的资源使用情况。要查看有关为 IAM 角色和其他用途进行标记的更多信息, 请访问了解更多链接。要继续, 请选择下一步: 审核。
6. 在审核页上, 根据需要进行以下更改:
  - 输入角色名称并 ( 可选 ) 更新说明。
  - 请验证以下内容:
    - 可信实体设置为 “AWS 服务: gamelift.amazonaws.com”。创建角色后, 必须更新此值。
    - 策略包括 GameLiftGameServerGroupPolicy。

要完成任务, 请选择创建角色。

7. 创建新角色后, 您必须手动更新角色的信任关系。转到角色页面并选择新角色名称以打开其摘要页面。打开信任关系选项卡, 然后选择编辑信任关系。在策略文档中, 更新 Service 属性以包含 autoscaling.amazonaws.com。修订后的 Service 属性应如下所示:

```
"Service": [  
  "gamelift.amazonaws.com",  
  "autoscaling.amazonaws.com"  
]
```

要保存更改, 请选择更新信任策略。

该角色已准备就绪。记下角色的 ARN 值, 该值显示在角色摘要页面的顶部。在设置 Amazon GameLift FleetIQ 游戏服务器组时, 您将需要这些信息。

## AWS CLI

这些步骤描述了如何 GameLift 使用 AWS CLI 为 Amazon 创建具有托管策略的服务角色。

1. 使用以下 JSON 语法创建信任策略文件 ( 示例 : FleetIQtrustpolicyGameLift.json )。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "gamelift.amazonaws.com",
          "autoscaling.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. 使用 [iam create-role](#) 创建一个新的 IAM 角色，并将其与您刚创建的信任策略 JSON 文件关联。

Windows:

```
AWS iam create-role --role-name FleetIQ-role-for-GameLift --assume-role-policy-document file://C:\policies\FleetIQtrustpolicyGameLift.json
```

Linux :

```
AWS iam create-role --role-name FleetIQ-role-for-GameLift --assume-role-policy-document file://policies/FleetIQtrustpolicyGameLift.json
```

请求成功时，响应将包括新创建的角色属性。记下 ARN 值。在设置 Amazon GameLift FleetIQ 游戏服务器组时，您将需要这些信息。

3. 使用 [iam 附加托管权限策略 “GameLiftGameServerGroupPolicy”](#)。 [attach-role-policy](#)

```
AWS iam attach-role-policy --role-name FleetIQ-role-for-GameLift --policy-arn arn:aws:iam::aws:policy/GameLiftGameServerGroupPolicy
```

要验证权限策略是否已附加，请使用新[角色 list-attached-role-policies](#)的名称调用 iam。

该角色已准备就绪。您可以通过调用 [gamelift create-game-server-group](#) 来验证 IAM 角色的配置是否正确，并将 `role-arn` 属性设置为新角色的 ARN 值。当 `GameServerGroup` 进入活动状态时，这表明 Amazon GameLift FleetsIQ 能够按预期修改您账户中的 Amazon EC2 和 Auto Scaling 资源。

## 为 Amazon EC2 创建角色

此角色使您的亚马逊 EC2 资源能够与亚马逊 GameLift FleetsIQ 进行通信。例如，在 Amazon EC2 实例上运行的游戏服务器需要能够报告运行状况。创建 Amazon GameLift FleetsIQ 游戏服务器组时，将此角色与您的 Amazon EC2 启动模板一起包含在 IAM 实例配置文件中。

使用 AWS CLI 为 Amazon EC2 创建角色，附加具有必要权限的自定义策略，并将该角色附加到实例配置文件。有关更多信息，请参阅 [AWS 服务创建角色](#)。

### AWS CLI

以下步骤介绍如何使用为 Amazon EC2 创建具有自定义 Amazon GameLift 权限的服务角色 AWS CLI。

1. 使用以下 JSON 语法创建信任策略文件（示例：`FleetsIQtrustpolicyEC2.json`）。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. 使用 [iam create-role](#) 创建一个新的 IAM 角色，并将其与您刚创建的信任策略 JSON 文件关联。

Windows:

```
AWS iam create-role --role-name FleetsIQ-role-for-EC2 --assume-role-policy-document file:///C:\policies\FleetsIQtrustpolicyEC2.json
```



Linux :

```
AWS iam create-role --role-name FleetIQ-role-for-EC2 --assume-role-policy-document file://policies/FleetIQtrustpolicyEC2.json
```

请求成功时，响应将包括新创建的角色属性。记下 ARN 值。您在设置 Amazon EC2 启动模板时需要此信息。

3. 使用以下 JSON 语法创建权限策略文件（示例：FleetIQpermissionsEC2.json）。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "gamelift:*",
      "Resource": "*"
    }
  ]
}
```

4. [使用 iam put-role-policy 将您刚刚创建的权限策略 JSON 文件附加到新角色。](#)

Windows:

```
AWS iam put-role-policy --role-name FleetIQ-role-for-EC2 --policy-name FleetIQ-permissions-for-EC2 --policy-document file://C:\policies\FleetIQpermissionsEC2.json
```

Linux :

```
AWS iam put-role-policy --role-name FleetIQ-role-for-EC2 --policy-name FleetIQ-permissions-for-EC2 --policy-document file://policies/FleetIQpermissionsEC2.json
```

要验证权限策略是否已附加，请使用新[角色 list-role-policies](#)的名称调用 iam。

5. 使用 iam 创建带有新角色 create-instance-profile 的实例配置文件以用于 [Amazon EC2](#)。有关更多信息，请参阅[管理实例配置文件](#)。

```
AWS iam create-instance-profile --instance-profile-name FleetIQ-role-for-EC2
```

当请求成功时，响应将包括新创建的实例配置文件的属性。

6. 使用 [add-role-to-instanceiam-profile](#) 将角色附加到实例配置文件。

```
AWS iam add-role-to-instance-profile --role-name FleetIQ-role-for-EC2 --  
instance-profile-name FleetIQ-role-for-EC2
```

此角色和配置文件现在可与 Amazon EC2 启动模板配合使用。

# 为 Amazon GameLift FleetIQ 准备游戏

本节介绍如何实现使用 Amazon GameLift FleetIQ 在 Amazon EC2 上托管游戏的设计。要启动和运行多人游戏，您需要执行以下操作：

- 调整您的游戏服务器以与 Amazon GameLift FleetIQ 通信。
- 创建一个 FleetIQ 游戏服务器组来部署游戏服务器。
- 向游戏客户端服务添加功能以请求可用的游戏服务器。

本节中的主题介绍如何完成这项工作的详细信息。要开始使用，请参阅集成计划，其中提供了详细的分步指南。

## 主题

- [Amazon GameLift FleetIQ 集成步骤](#)
- [管理 Amazon GameLift FleetIQ 游戏服务器组](#)
- [将 Amazon GameLift FleetIQ 集成到游戏服务器中](#)
- [将 Amazon GameLift FleetIQ 集成到游戏客户端中](#)

## Amazon GameLift FleetIQ 集成步骤

该集成计划概述了使用 Amazon GameLift FleetIQ 在 Amazon EC2 实例上启动和运行多人游戏的关键步骤。如果您正在寻找可自动执行更多游戏托管流程的 Amazon GameLift 托管服务，请参阅 [Amazon GameLift 开发人员指南](#)。

要开始使用 Amazon GameLift FleetIQ，您应该有一个在本地环境或 Amazon EC2 环境中正常运行的游戏服务器。游戏服务器可以是管理一个或多个游戏会话、生成子进程或在容器内运行的单个进程。

1. 注册一个 [AWS 账户](#) 并设置用户的 Amazon GameLift FleetIQ 访问权限。

创建一个新 AWS 账户 账户或选择一个现有账户来使用 Amazon GameLift FleetIQ。为用户设置管理游戏中使用的 Amazon EC2、Auto Scaling 和其他 AWS 资源的权限。有关详细说明，请参阅 [为亚马逊 GameLift FleetIQ 设置 AWS 账户](#)。

2. 创建 IAM 角色。

创建允许 Amazon GameLift FleetIQ、Amazon EC2 和 Auto Scaling 资源相互通信的角色。有关更多信息，请参阅 [为跨服务交互创建 IAM 角色](#)。

### 3. 使用 Amazon GameLift FleetIQ 功能获取 AWS 软件开发工具包和 AWS CLI。

- [下载最新版本的 AWS 软件开发工具包](#)。
- [查看 Amazon GameLift API 参考文档](#)。

### 4. 准备好您的游戏服务器，以便在 Amazon GameLift FleetIQ 上使用。

将 AWS 软件开发工具包添加到您的游戏服务器项目中，并添加代码，让 Amazon GameLift FleetIQ 随时了解游戏服务器的当前状态和使用情况。有关其他指导和示例，请参阅[the section called “集成游戏服务器”](#)。Amazon GameLift FleetIQ 使用这些信息为您的对战系统提供可行、未占用的游戏服务器列表，还可以避免在平衡期间终止当前托管玩家的实例。

### 5. 使用您的游戏服务器创建 Amazon EC2 亚马逊机器映像 (AMI)。

使用您的游戏服务器软件以及任何其他运行时资产或配置设置创建 AMI。如需帮助，请参阅《Amazon EC2 用户指南》中的[亚马逊机器映像 \(AMI\)](#)。

### 6. 创建 Amazon EC2 启动模板。

构建 Amazon EC2 启动模板，该模板使用您的自定义 AMI，并为您的托管资源定义网络和安全设置。启动模板必须引用您创建的实例配置文件（请参阅步骤 2），并具有允许您的游戏服务器与 Amazon GameLift FleetIQ 通信的权限。您无需在启动模板中包含实例类型，因为这将在稍后完成。如需帮助，请参阅《Amazon EC2 用户指南》中的[创建启动模板](#)。

#### Note

在将启动模板与 Amazon GameLift FleetIQ 结合使用之前，我们强烈建议您先设置一个自动扩缩组，以验证模板配置和 AMI 是否正确部署。

### 7. 设置 Amazon GameLift FleetIQ 托管资源。

在要部署游戏服务器的每个区域中，通过调用 [CreateGameServerGroup\(\)](#) 来创建游戏服务器组。传入启动模板（包含您的自定义 AMI 以及网络和安全设置）、IAM 角色以及游戏可以运行的实例类型列表。此操作会在您的 AWS 账户中设置一个自动扩缩组，Amazon GameLift FleetIQ 可以修改该组。有关其他指导和示例，请参阅[管理 Amazon GameLift FleetIQ 游戏服务器组](#)。

### 8. 将 Amazon GameLift FleetIQ 集成到游戏客户端中。

将 AWS 软件开发工具包添加到您的游戏客户端、对战构建器或分配游戏服务器容量的其他后端组件中。根据您的游戏类型，对战构建器可能会调用 [ListGameServers\(\)](#) 或 [ClaimGameServer\(\)](#) 以获取服务器容量并预留可用的游戏服务器。有关其他指导和示例，请参阅[将 Amazon GameLift FleetIQ 集成到游戏客户端中](#)。

## 9. 扩展您的自动扩缩组。

在您的自动扩缩组中预配置实例时，它们会启动您的游戏服务器。然后，每个游戏服务器都注册到 Amazon GameLift FleetIQ 作为可用容量，以便稍后在您的对战构建器中列出或认领。

## 10 测试游戏。

调用您的对战构建器并调用 `ClaimGameServer` 以请求服务器容量。将生成的 IP 和端口传回游戏客户端，以便他们可以连接到游戏服务器。

# 管理 Amazon GameLift FleetIQ 游戏服务器组

本主题介绍设置 Amazon GameLift FleetIQ 游戏服务器组所需的任务。创建游戏服务器组会触发一个 EC2 自动扩缩组的创建，该组具有所有必要的配置设置以及用于管理游戏托管的 Amazon GameLift FleetIQ 优化的配置。

在创建游戏服务器组之前，您必须至少准备好以下资源：

- 一个 Amazon EC2 启动模板，用于指定如何使用游戏服务器构建启动 Amazon EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[通过启动模板启动实例](#)。
- 一个 IAM 角色，可扩展对 AWS 的有限访问权限，以允许 GameLift FleetIQ 创建自动扩缩组并与之交互。有关更多信息，请参阅[为跨服务交互创建 IAM 角色](#)。

## 创建游戏服务器组

要创建游戏服务器组，请调用 [CreateGameServerGroup\(\)](#)。此操作会同时创建 Amazon GameLift FleetIQ 游戏服务器组和相应自动扩缩组。创建游戏服务器组时，您需要为 Amazon GameLift FleetIQ 提供特定于游戏的设置，包括平衡策略和实例类型定义。您还为自动扩缩组提供初始属性设置。

以下示例触发创建 `GameServerGroup`，其指定 `c4.large` 和 `c5.large` 实例类型并将该组限制为仅限竞价型实例，以及一个自动扩缩组，该组使用指定的启动模板部署实例，并使用目标跟踪自动扩展策略在最小和最大设置范围内管理组容量。短暂的预配置时段后，将创建一个 `AutoScalingGroup` 资源，并且 `GameServerGroup` 进入 `ACTIVE` 状态。

```
AWS gamelift create-game-server-group \  
  --game-server-group-name MyLiveGroup \  
  --role-arn arn:aws:iam::123456789012:role/GameLiftGSGRole \  
  --min-size 1 \  
  --max-size 10 \  
  --target-tracking-enabled true \  
  --target-tracking-min-size 1 \  
  --target-tracking-max-size 10 \  
  --target-tracking-cooldown-period 300 \  
  --target-tracking-cooldown-type TargetTypeCooldown
```

```
--game-server-protection-policy FULL_PROTECTION \  
--balancing-strategy SPOT_ONLY \  
--launch-template LaunchTemplateId=lt-012ab345cde6789ff \  
--instance-definitions '[{"InstanceType": "c4.large"}, {"InstanceType":  
"c5.large"}]' \  
--auto-scaling-policy '{"TargetTrackingConfiguration": {"TargetValue": 66}}'
```

## 更新游戏服务器组

您可以更新会影响 Amazon GameLift FleetIQ 管理游戏服务器托管方式的游戏服务器组属性，包括资源类型优化。若要更新这些属性，调用 [UpdateGameServerGroup\(\)](#)。对游戏服务器组的更改生效后，Amazon GameLift FleetIQ 可能会覆盖自动扩缩组中的某些属性。

对于所有其他自动扩缩组属性，例如 MinSize、MaxSize、和 LaunchTemplate，您可以直接在自动扩缩组上修改这些属性。

在下面的示例中，更新实例类型定义以切换到 c4.xlarge 和 c5.xlarge 实例类型。

```
AWS gamelift update-game-server-group \  
--game-server-group-name MyLiveGroup \  
--instance-definitions '[{"InstanceType": "c4.xlarge"}, {"InstanceType":  
"c5.xlarge"}]'
```

## 跟踪游戏服务器组实例

在创建实例并将其部署到您的游戏服务器组和自动扩缩组之后，您可以通过调用 [DescribeGameServerInstances\(\)](#) 来跟踪游戏服务器实例的状态。您可以使用此操作来跟踪实例状态。有关游戏服务器组状态的更多信息，请参阅[游戏服务器组的生命周期](#)。

您还可以使用游戏服务器组下的 [Amazon GameLift 控制台](#) 来监控您的游戏服务器组的状态。

## 将 Amazon GameLift FleetIQ 集成到游戏服务器中

本主题介绍准备游戏服务器项目以与 Amazon GameLift FleetIQ 通信所需的任务。有关其他指导，请参阅[Amazon GameLift FleetIQ 最佳实操](#)。

## 注册游戏服务器

当游戏服务器进程启动并准备好托管直播游戏时，它必须通过调用 [RegisterGameServer\(\)](#) 向 Amazon GameLift FleetIQ 注册。注册后，Amazon GameLift FleetIQ 可以在对战系统或其他客户服务请求服务

器容量信息或申领游戏服务器时对对战系统或其他客户端服务做出响应。注册时，游戏服务器可以向 Amazon GameLift FleetIQ 提供相关的游戏服务器数据和连接信息，包括它用于入站客户端连接的端口和 IP 地址。

```
AWS gamelift register-game-server \  
  --game-server-id UniqueId-1234 \  
  --game-server-group-name MyLiveGroup \  
  --instance-id i-1234567890 \  
  --connection-info "1.2.3.4:123" \  
  --game-server-data "{\"key\": \"value\"}"
```

## 更新游戏服务器状态

游戏服务器注册后，应定期报告运行状况和利用率状态，以便在 Amazon GameLift FleetIQ 上保持服务器容量状态同步。您可以通过调用 [UpdateGameServer\(\)](#) 报告运行状况和利用状态。在下面的示例中，游戏服务器报告正常运行，并且未由玩家或游戏占用。

```
AWS gamelift update-game-server \  
  --game-server-group-name MyLiveGroup \  
  --game-server-id UniqueId-1234 \  
  --health-check HEALTHY \  
  --utilization-status AVAILABLE
```

### 运行状况

如果您的游戏服务器具有跟踪运行状况状态的机制，则可以使用此机制来触发游戏服务器对 Amazon GameLift FleetIQ 的运行状况更新。

### 利用率状态

报告游戏服务器利用率状态可以让 Amazon GameLift FleetIQ 了解哪些游戏服务器目前最适合用于新游戏会话。您的游戏服务器必须具有触发 Amazon GameLift FleetIQ 利用率状态更新的机制。例如，当玩家连接到游戏服务器或游戏会话开始时，可能会触发更新。

开始游戏会话时，客户端或对战服务会话会声明可用的游戏服务器（通过调用 [ClaimGameServer\(\)](#)），提示玩家连接到游戏服务器，然后触发游戏服务器开始游戏。[将 Amazon GameLift FleetIQ 集成到游戏客户端中](#) 介绍了此过程。游戏服务器“声明”的有效期为 60 秒，并且游戏服务器必须能够在此窗口内更新利用率状态。如果未更新利用率状态，Amazon GameLift FleetIQ 会删除该声明，假设游戏服务器可用，并可能保留游戏服务器以供其他客户端声明请求。

```
AWS gamelift update-game-server \  
  --game-server-group-name MyLiveGroup \  
  --game-server-id UniqueId-1234 \  
  --health-check HEALTHY \  
  --utilization-status UTILIZED
```

## 注销游戏服务器

游戏结束后，游戏服务器必须使用 [DeregisterGameServer\(\)](#) 从 Amazon GameLift FleetIQ 注销。

```
AWS gamelift deregister-game-server \  
  --game-server-group-name MyLiveGroup \  
  --game-server-id UniqueId-1234
```

## 将 Amazon GameLift FleetIQ 集成到游戏客户端中

本主题描述了准备游戏客户端或对战服务以与 Amazon GameLift FleetIQ 通信以获取用于托管游戏会话的游戏服务器所需的任务。

创建一种方法，允许游戏客户端或对战构建器为玩家请求游戏服务器资源。为此，您可以在几种方法中选择：

- 让 Amazon GameLift FleetIQ 选择可用的游戏服务器。此选项利用 Amazon GameLift FleetIQ 优化，使用低成本竞价型实例并进行自动扩展。
- 请求所有可用的游戏服务器并选择一个以使用（通常称为“列出并挑选”）。

## 让 Amazon GameLift FleetIQ 选择一台游戏服务器

要让 Amazon GameLift FleetIQ 选择可用的游戏服务器，请在不指定游戏服务器 ID 的情况下调用 [ClaimGameServer\(\)](#)。在这种情况下，Amazon GameLift FleetIQ 确实运用其逻辑在实例上查找可用于游戏托管并且针对自动扩缩进行了优化的游戏服务器。

```
AWS gamelift claim-game-server \  
  --game-server-group-name MyLiveGroup
```

为响应认领请求，Amazon GameLift FleetIQ 标识 GameServer 资源、连接信息和游戏数据，客户端可使用这些数据来连接到游戏服务器。游戏服务器的认领状态设置为 CLAIMED 60 秒。玩家连接或



游戏开始后，您的游戏服务器或客户端服务需要在 Amazon GameLift FleetIQ 上更新游戏服务器的状态。此机制确保 Amazon GameLift FleetIQ 不在后续对游戏服务器容量请求的响应中提供此游戏服务器。通过调用 [UpdateGameServer\(\)](#) 来更新游戏服务器状态。

```
AWS gamelift update-game-server \  
  --game-server-group-name MyLiveGroup \  
  --game-server-id UniqueId-1234 \  
  --health-check HEALTHY \  
  --utilization-status UTILIZED
```

## 选择您自己的游戏服务器

要使用“列出并挑选”方法，请让您的游戏客户端或对战构建器调用 [ListGameServers\(\)](#) 来请求可用游戏服务器的列表。在选择游戏服务器时，您可能希望使用游戏服务器数据来提供玩家或对战构建器可以使用的其他信息。要控制结果的返回方式，您可以请求分页结果并按注册日期对游戏服务器进行排序。以下请求返回指定游戏服务器组中的 20 个处于活动状态且可用的游戏服务器，按注册时间排序，最新的游戏服务器排在最前面。

```
AWS gamelift list-game-servers \  
  --game-server-group-name MyLiveGroup \  
  --limit 20 \  
  --sort-order DESCENDING
```

根据可用游戏服务器的列表，客户端或对战服务选择一个游戏服务器，然后使用特定的游戏服务器 ID 调用 [ClaimGameServer\(\)](#) 来声明该服务器。在这种情况下，Amazon GameLift FleetIQ 不执行任何其实例类型优化逻辑，如[亚马逊 GameLift FleetIQ 逻辑](#)所述。

```
AWS gamelift claim-game-server \  
  --game-server-group-name MyLiveGroup \  
  --game-server-id UniqueId-1234
```

# 使用 Amazon CloudWatch 监控 Amazon GameLift FleetIQ

Amazon CloudWatch 指标可以帮助您扩展实例容量、构建操作控制面板并触发警报。Amazon GameLift FleetIQ 作为独立解决方案，会向您的 AWS 账户发布一组 Amazon CloudWatch 指标。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的[使用 Amazon CloudWatch 监控自动扩缩组和实例](#)。

此处列出了 FleetIQ 指标。在 [Amazon GameLift 指标](#) 中查看 Amazon GameLift 的完整 Amazon CloudWatch 指标信息。

指标	描述
AvailableGameServers	<p>可用于运行游戏执行但当前未被玩游戏占用的游戏服务器数量。此数字包括已认领但仍处于 AVAILABLE (可用) 状态的游戏服务器。</p> <p>单位：计数</p> <p>相关的 CloudWatch 统计数据：Sum</p> <p>维度：GameServerGroup</p>
UtilizedGameServers	<p>当前被游戏占用的游戏服务器数量。此数字包括处于 UTILIZED 状态的游戏服务器。</p> <p>单位：计数</p> <p>相关的 CloudWatch 统计数据：Sum</p> <p>维度：GameServerGroup</p>
DrainingAvailableGameServers	<p>当前不支持玩游戏且计划终止的实例上的游戏服务器数量。这些游戏服务器属于为响应新的认领请求而认领的最低优先级。</p> <p>单位：计数</p> <p>相关的 CloudWatch 统计数据：Sum</p> <p>维度：GameServerGroup</p>

指标	描述
DrainingUtilizedGameServers	<p>当前支持玩游戏且计划终止的实例上的游戏服务器数量。</p> <p>单位：计数</p> <p>相关的 CloudWatch 统计数据：Sum</p> <p>维度：GameServerGroup</p>
PercentUtilizedGameServers	<p>当前支持游戏执行的游戏服务器所占的部分。此指标表示当前正在使用的游戏服务器容量。它可用于促使制定相应的自动扩缩策略，以便可以动态添加和删除实例来与玩家需求匹配。</p> <p>单位：百分比</p> <p>相关的 CloudWatch 统计数据：Average、Minimum、Maximum</p> <p>维度：GameServerGroup</p>
GameServerInterruptions	<p>由于竞价型实例可用性有限而中断的竞价型实例上的游戏服务器数量。</p> <p>单位：计数</p> <p>相关的 CloudWatch 统计数据：Sum</p> <p>维度：GameServerGroup、InstanceType</p>
InstanceInterruptions	<p>由于可用性有限而中断的竞价型实例数量。</p> <p>单位：计数</p> <p>相关的 CloudWatch 统计数据：Sum</p> <p>维度：GameServerGroup、InstanceType</p>

# Amazon GameLift FleetIQ 参考指南

本节包含用于 Amazon GameLift FleetIQ 的参考文档。

## 主题

- [Amazon GameLift FleetIQ 服务 API 参考 \( AWS 软件开发工具包 \)](#)

## Amazon GameLift FleetIQ 服务 API 参考 ( AWS 软件开发工具包 )

本主题提供了适用于 Amazon GameLift FleetIQ 的 API 操作的基于任务的列表。Amazon GameLift FleetIQ 服务 API 已打包到 `aws.gamelift` 命名空间中的 AWS 软件开发工具包中。 [下载 AWS 软件开发工具包](#) 或 [查看 Amazon GameLift API 参考文档](#)。

Amazon GameLift FleetIQ 优化低成本竞价型实例的使用，通过 Amazon EC2 进行基于云的游戏托管。有关其他 Amazon GameLift 托管选项的更多信息，请参阅 [Amazon GameLift 开发人员指南](#)。

## Amazon GameLift FleetIQ API 操作

以下操作允许您与 Amazon EC2 和自动扩缩组一起管理您的 Amazon GameLift FleetIQ 资源，包括游戏服务器组和游戏服务器。

### 管理游戏服务器组

使用这些操作通过 FleetIQ 优化来管理您的游戏服务器部署。游戏服务器组控制游戏服务器进程在 Amazon EC2 实例上的启动方式、设置和自动扩缩组，并定义如何应用 FleetIQ 优化。

- [CreateGameServerGroup](#) – 创建新的游戏服务器组和对应的自动扩缩组，然后开始启动实例以托管游戏服务器。CLI 命令：[create-game-server-group](#)
- [ListGameServerGroups](#) – 获取 Amazon GameLift 区域中所有游戏服务器组的列表。CLI 命令：[list-game-server-groups](#)
- [DescribeGameServerGroup](#) – 检索游戏服务器组的元数据。CLI 命令：[describe-game-server-group](#)
- [UpdateGameServerGroup](#) – 更改游戏服务器组元数据。CLI 命令：[update-game-server-group](#)
- [DeleteGameServerGroup](#) – 永久删除游戏服务器组并终止关联托管资源的 FleetIQ 活动。CLI 命令：[delete-game-server-group](#)
- [ResumeGameServerGroup](#) – 恢复游戏服务器组暂停的 FleetIQ 活动。CLI 命令：[resume-game-server-group](#)

- [SuspendGameServerGroup](#) – 临时停止游戏服务器组的 FleetIQ 活动。CLI 命令：[suspend-game-server-group](#)

## 管理游戏服务器

使用这些操作通过 FleetIQ 优化来管理您的游戏服务器部署。游戏服务器组控制游戏服务器进程在 Amazon EC2 实例上的启动方式、设置和自动扩缩组，并定义如何应用 FleetIQ 优化。

- [RegisterGameServer](#) – 从新的游戏服务器进行调用，以通知 Amazon GameLift FleetIQ 游戏服务器已准备好托管游戏。CLI 命令：[register-game-server-group](#)
- [ListGameServers](#) – 从游戏客户端服务进行调用，以获取当前在游戏服务器组中运行的所有游戏服务器的列表。CLI 命令：[list-game-servers](#)
- [ClaimGameServer](#) – 从游戏客户端服务进行调用，以查找并保留游戏服务器以托管新游戏会话。CLI 命令：[claim-game-server](#)
- [DescribeGameServer](#) – 检索游戏服务器的元数据。CLI 命令：[describe-game-server](#)
- [UpdateGameServer](#) – 更改游戏服务器元数据、运行状况或利用率状态。CLI 命令：[update-game-server](#)
- [DeregisterGameServer](#) – 从终止游戏服务器进行调用，以提示 Amazon GameLift FleetIQ 从游戏服务器组中删除游戏服务器。CLI 命令：[deregister-game-server](#)

## 可用编程语言

支持 Amazon GameLift 的 AWS 软件开发工具包有以下语言版本。有关开发环境支持的信息，请参阅每种语言的文档。

- C++ ( [软件开发工具包文档](#) ) ( [Amazon GameLift](#) )
- Java ( [软件开发工具包文档](#) ) ( [Amazon GameLift](#) )
- .NET ( [软件开发工具包文档](#) ) ( [Amazon GameLift](#) )
- Go ( [软件开发工具包文档](#) ) ( [Amazon GameLift](#) )
- Python ( [软件开发工具包文档](#) ) ( [Amazon GameLift](#) )
- Ruby ( [软件开发工具包文档](#) ) ( [Amazon GameLift](#) )
- PHP ( [软件开发工具包文档](#) ) ( [Amazon GameLift](#) )
- JavaScript/Node.js ( [软件开发工具包文档](#) ) ( [Amazon GameLift](#) )

# 亚马逊 GameLift FleetiQ 的安全

如果您将 Amazon GameLift FleetiQ 用作亚马逊 EC2 的独立功能，[请参阅亚马逊 EC2 用户指南中的亚马逊 EC 2 中的安全](#)。

云安全 AWS 是重中之重。为了满足对安全性最敏感的组织的需求，我们打造了具有超高安全性的数据中心和网络架构。作为 AWS 的客户，您也可以从这些数据中心和网络架构受益。

安全是双方共同承担 AWS 的责任。有关在使用 Amazon GameLift FleetiQ 时如何应用分担责任模式的信息，[请参阅亚马逊安全](#)。 GameLift

# Amazon GameLift FleetIQ 发行说明和软件开发工具包版本

Amazon GameLift 发行说明提供与服务相关的新 FleetIQ 特征、更新和修复的详细信息。本页还包括 Amazon GameLift 软件开发工具包版本历史记录。

# Amazon GameLift 开发人员资源

要查看所有 Amazon GameLift 文档和开发人员资源，请参阅 [Amazon GameLift 文档](#) 主页。



# AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 [AWS 词汇表](#)。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。