



FleetIQ Developer Guide

Amazon GameLift Servers



Version

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon GameLift Servers: FleetIQ Developer Guide

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していない他のすべての商標は、それぞれの所有者の所有物であり、Amazon と提携、接続、または後援されている場合とされていない場合があります。

Table of Contents

とは Amazon GameLift ServersFleetIQ	1
FleetIQ の働き	2
Amazon GameLift ServersFleetIQ ロジック	2
主なリソースとコンポーネント	6
ゲームアーキテクチャ	8
オンプレミスホスティングの補完	8
ゲーム サーバーグループの寿命	10
ゲームサーバーの寿命	12
スポット再調整プロセス	15
ベストプラクティス	17
Amazon GameLift ServersfeaturesFleetIQ	20
Amazon GameLift ServersFleetIQ の料金	21
セットアップ	22
対応ソフトウェア	22
AWS アカウントをセットアップする	23
を作成する AWS アカウント	23
のユーザーアクセス許可を管理する Amazon GameLift ServersFleetIQ	26
サービス間でやり取りするための IAM ロールの作成	30
のゲームの準備 FleetIQ	37
統合ステップ	37
ゲーム サーバーグループの管理	39
ゲーム サーバーグループを作成する	40
ゲーム サーバーグループを更新する	41
ゲーム サーバーグループインスタンスの追跡	41
ゲームサーバーの統合	41
ゲームサーバーを登録する	42
ゲームサーバーのステータスを更新する	42
ゲームサーバーの登録を解除する	43
ゲームクライアントの統合	43
ゲームサーバーAmazon GameLift ServersFleetIQを選択	44
自分のゲームサーバーを選択する	45
CloudWatch を使用して監視する	46
FleetIQ でのセキュリティ	49
Amazon GameLift Servers FleetIQ リファレンス	50

サービス API リファレンス (AWS SDK)	50
Amazon GameLift ServersFleetIQ API アクション	50
利用可能なプログラミング言語	52
リリースノートと SDK バージョン	52
すべてのAmazon GameLift Serversガイド	53
AWS 用語集	54
.....	iv

とは Amazon GameLift ServersFleetIQ

Amazon GameLift ServersFleetIQ は、クラウドベースのゲームホスティングのための低コストの Amazon Elastic Compute Cloud (Amazon EC2) スポットインスタンスの使用を最適化します。を使用すると Amazon GameLift ServersFleetIQ、最適化を活用して低コストで Amazon EC2 および Amazon EC2 Auto Scaling のホスティングリソースを直接操作できます。Amazon GameLift Servers Amazon EC2 スポットインスタンスは、大幅な割引で提供されますが、可用性が変動し、[中断](#)が発生する可能性があるため、ゲームホスティングでは一般的に使用できません。これらの制限 Amazon GameLift ServersFleetIQ を大幅に軽減し、低コストのスポットインスタンスをゲームホスティングで使用できます。

FleetIQ 最適化は、Amazon GameLift Servers を使用してゲームホスティングを管理する場合にも利用できます。Amazon GameLift Servers ホスティングオプションの詳細については、[Amazon GameLift Servers デベロッパーガイド](#) を参照してください。

Amazon GameLift ServersFleetIQ ゲームホスティングソリューションは、以下のゲーム開発者向けに設計されています。

- 既存の AWS デプロイがある場合、またはフルマネージド Amazon GameLift Servers サービスではなく Amazon EC2 を直接使用したい場合。Amazon GameLift ServersFleetIQ は、で管理する EC2 Auto Scaling グループと連携して AWS アカウント、EC2 インスタンスとグループへのフルアクセスを提供します。Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS)、などの他の AWS サービスと統合することもできます AWS Shield Advanced。
- 既存のオンプレミスのゲームホスティングがあり、容量をクラウドに拡張したいユーザー。を使用すると Amazon GameLift ServersFleetIQ、オンプレミスの容量を使用し、必要に応じて AWS クラウド容量を段階的に追加するハイブリッドデプロイシステムを構築できます。

の使用を開始する準備はできました Amazon GameLift ServersFleetIQ か？

- スキルビルダー—Amazon GameLift ServersFleetIQ で [Amazon Amazon GameLift ServersFleetIQ for Game Servers を使用する](#) コースを受講して、ゲームに AWS を使用する方法について説明します。関連コースの概要については、「[ゲームテック学習プラン](#)」を参照してください。一部のクラスはさまざまな言語で提供されています。
- 「[Amazon GameLift ServersFleetIQ 統合ステップ](#)」の手順に従います。

のAmazon GameLift ServersFleetIQ仕組み

このAmazon GameLift ServersFleetIQソリューションは、Amazon EC2 と Auto Scaling で利用できるコンピューティングリソース管理ツールの完全なセットを補完するゲームホスティングレイヤーです。ゲームホスティングに固有の機能のスレートを提供するだけでなく、Amazon GameLift ServersFleetIQは、ゲームホスティングに低コストのスポットインスタンスを使用できるようにするロジックのレイヤーを提供します。このソリューションを使用すると、Amazon EC2 および Auto Scaling リソースを直接管理し、必要に応じて他の AWS のサービスと統合できます。

を使用する場合はAmazon GameLift ServersFleetIQ、通常どおり Amazon EC2 インスタンスを起動する準備をします。ゲームサーバーソフトウェアで Amazon マシンイメージ (AMI) を作成し、Amazon EC2 起動テンプレートを作成し、Auto Scaling グループの設定を定義します。ただし、Auto Scaling グループを直接作成する代わりに、Amazon EC2 と Auto Scaling のリソースと設定を使用してAmazon GameLift ServersFleetIQゲームサーバーグループを作成します。このアクションではAmazon GameLift ServersFleetIQ、ゲームサーバーグループと対応する Auto Scaling グループの両方を作成するように求められます。ゲームサーバーグループは、Auto Scaling グループの特定の側面にリンクされ、管理されます。

Auto Scaling グループが作成されると、Amazon EC2 リソースと Auto Scaling リソースへのフルアクセス権が付与されます。Auto Scaling グループの設定を変更したり、マルチレベルスケールリングポリシーやロードバランサーを追加したり、他の AWS サービスと統合したりできます。グループ内のインスタンスに直接Connectできます。最適化ロジックの一部として、は特定の Auto Scaling グループプロパティ Amazon GameLift ServersFleetIQを定期的に更新します。Auto Scaling グループによってデプロイされたすべてのインスタンスの可用性ステータスを追跡できます。

ゲームサーバーグループのAmazon GameLift ServersFleetIQアクティビティはいつでも一時的に停止できます。また、ゲームスケールリンググループを削除しても、対応する Auto Scaling グループは保持できるオプションがあります。

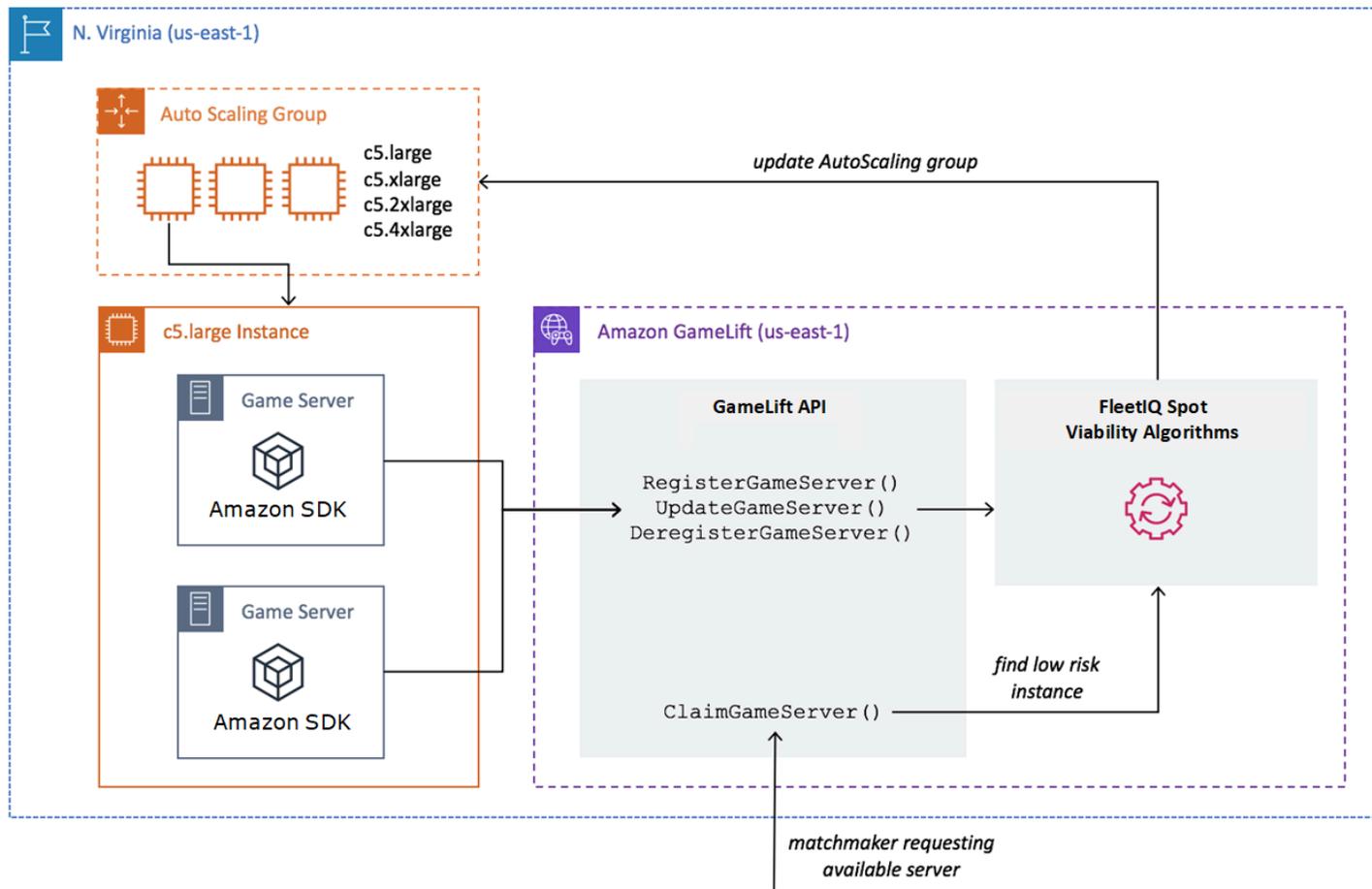
トピック

- [Amazon GameLift ServersFleetIQ ロジック](#)
- [主なリソースとコンポーネント](#)

Amazon GameLift ServersFleetIQ ロジック

次の図Amazon GameLift ServersFleetIQは、ゲームホスティングで Amazon EC2 を使用する場合の の役割を示しています。その主な目標は、ゲームセッションをホストし、プレイヤーに最適な

ゲームプレイエクスペリエンスを提供する最適なゲームサーバーを見つけることです。は、最も低コストで最高のゲームホスティングの実行可能性を提供するものとして最適なリソースAmazon GameLift ServersFleetIQを定義します。は、この目標に 2 つの主要な方法でAmazon GameLift ServersFleetIQアプローチします。1 つ目は Auto Scaling グループで実行可能なインスタンスタイプのみを許可し、2 つ目は、グループの利用可能なリソース全体に新しいゲームセッションを効果的に配置することです。



Auto Scaling グループに最適なインスタンスタイプを設定する

Auto Scaling グループのジョブは、新しいインスタンスを起動して古いインスタンスを使用停止することで、ホスティングリソースのコレクションを維持し、これをプレイヤーの需要に合わせてスケールリングすることです。これを行うには、Auto Scaling グループは目的のインスタンスタイプのリストに依存します。のジョブAmazon GameLift ServersFleetIQは、これらの必要なインスタンスタイプの実行可能性を継続的にチェックし、Auto Scaling グループのリストを更新することです。このプロセスは、インスタンスバランシングと呼ばれています。これにより、Auto Scaling グループ内のインスタンスが継続的に更新され、現在実行可能なインスタンスタイプのみが常に使用されるようになります。

Amazon GameLift Servers FleetIQ Auto Scaling グループが最適なインスタンスタイプを選択する方法には、次の方法があります。

- [It determines usage of Spot and/or On-Demand Instances].(スポットやオンデマンドインスタンスの利用状況を判断します)。Amazon GameLift Servers FleetIQ ゲームサーバーグループは、Auto Scaling グループがスポットインスタンスやオンデマンドインスタンスを使用する方法に影響するバランシング戦略で設定されます。スポットインスタンスは、可用性の変動や[中断](#)の可能性、ゲームサーバーのホスティング Amazon GameLift Servers FleetIQ を最小限に抑える制限により、コストが低くなります。オンデマンドインスタンスのコストは高くなりますが、必要なときにより信頼性の高い可用性を提供します。
- これにより、新しいインスタンスが実行可能なインスタンスタイプでのみ起動するように制限されます。Amazon GameLift Servers FleetIQ ゲームサーバーグループは、目的のインスタンスタイプのマスターリストを保持します。インスタンスバランシングプロセスは、インスタンスタイプの最近の可用性と中断率を調べる予測アルゴリズムを使用して、ゲームホスティングの実行可能性についてリスト内の目的の各インスタンスタイプを継続的に評価します。この評価の結果として、は Auto Scaling グループの必要なインスタンスタイプのリスト Amazon GameLift Servers FleetIQ を継続的に更新し、現在有効なインスタンスタイプのみを含めます。
- 実行不可能なインスタンスタイプである既存のインスタンスにフラグを付けます。は、現在実行不可能なインスタンスタイプである Auto Scaling グループ内の既存のインスタンス Amazon GameLift Servers FleetIQ を識別します。該当するインスタンスには、[draining]ドレイン中のフラグが付けられます。つまり、このようなインスタンスは終了され、新しいインスタンスに置き換えられます。ゲームサーバーの保護がオンになっているインスタンスの場合、アクティブなゲームセッションが正常に終了するまで終了が延期されます。

Auto Scaling グループは、インスタンスを起動して終了することにより、低コストのスポットインスタンスタイプの可用性が変動しても、ゲームホスティング用に最適化されたコレクションを維持します。バランシングアクティビティは、アクティブなインスタンスを持つゲームサーバーグループでのみ行われます。このプロセスの詳細については、「[スポット再調整プロセス](#)」を参照してください。

ゲームセッションを効果的に配置する

Amazon GameLift Servers FleetIQ は、ゲームサーバーグループ内のすべてのアクティブなゲームサーバーを追跡し、この情報を使用して新しいゲームセッションとプレイヤーに最適な配置を決定します。

Amazon GameLift ServersFleetIQ がゲームサーバーを追跡できるようにするには、ゲームサーバーソフトウェアがそのステータスをレポートする必要があります。カスタム AMI は、各インスタンスでの新しいゲームサーバープロセスの開始と停止の方法を管理します。新しいゲームサーバーが開始されると、 に登録され Amazon GameLift ServersFleetIQ、ゲームセッションをホストする準備ができていることを示します。登録後、ゲームサーバーは、その状態および現在ゲームセッションをホスティングしているかどうかを定期的にレポートします。ゲームサーバーがシャットダウンすると、への登録が解除されます Amazon GameLift ServersFleetIQ。

新しいゲームセッションを開始するには、ゲームクライアント (またはマッチメーカーやその他のクライアントサービス) からゲームサーバーへのリクエストが に送信されます Amazon GameLift ServersFleetIQ。 は、利用可能なゲームサーバー Amazon GameLift ServersFleetIQ を見つけ、新しいゲームセッションに対して要求し、ゲームサーバー ID と接続情報で応答します。これに応じて、ゲームサーバーはそのステータスを更新し、参加プレイヤーの新しいゲームセッションを開始します。

新しいゲームセッションをホストするゲームサーバーを選択する場合、 は次の意思決定プロセス Amazon GameLift ServersFleetIQ を使用して、実行可能な低コストのスポットインスタンスによる配置を最適化します。

1. 可能な場合、 は他のゲームセッションをすでにホストしているインスタンスに新しいゲームセッション Amazon GameLift ServersFleetIQ を配置します。一部のインスタンスに集中させて (ただし、過負荷にはせず)、他のインスタンスをアイドル状態に保つことで、Auto Scaling グループは不要なアイドル状態のインスタンスを迅速にスケールダウンできるため、ホスティングコストが削減されます。
2. Amazon GameLift ServersFleetIQ は、ドレーニングとしてフラグが付けられたインスタンス、つまりゲームホスティングでは実行できないインスタンスを無視します。これらのインスタンスは、既存のゲームセッションをサポートするためにだけに実行されています。これらは、他のゲームサーバーが利用できない限り、新しいゲームセッションには使用できません。
3. Amazon GameLift ServersFleetIQ は、有効なインスタンスで実行されているすべての利用可能なゲームサーバーを識別します。

ゲームサーバーグループのゲームセッション保護をオンにすると、アクティブに実行中のゲームセッションで Auto Scaling グループがインスタンスを終了するのを防ぐことができます。

主なリソースとコンポーネント

でゲームホスティングリソースを設定する前に、AWS アカウントに次のリソースを作成します Amazon GameLift ServersFleetIQ。ベストプラクティスとして、ゲームサーバーグループで使用する前に、これらのリソースを使用してゲームサーバーのデプロイを開発してテストします。

- Amazon マシンイメージ (AMI)。AMI は、Amazon EC2 インスタンスで起動する特定のソフトウェア設定のテンプレートです。ゲームホスティングの場合、AMI には、オペレーティングシステム、ゲームサーバーのバイナリまたはコンテナ、およびゲームサーバーに必要なその他のランタイムソフトウェアが含まれます。AMI 作成の詳細については、「Amazon EC2 ユーザーガイド」の「[Amazon マシンイメージ](#)」を参照してください。AMI はリージョン固有です。リージョン間でのAMI をコピーについては、[Amazon EC2 User Guide] (Amazon EC2 ユーザーガイド) の[\[Copying AMIs\]](#) (AMI のコピー) を参照してください。
- [Amazon EC2 launch template] (Amazon EC2 起動テンプレート) 起動テンプレートは、Auto Scaling グループ内のインスタンスを起動および管理するための手順を提供します。AMI を指定して、適切なインスタンスタイプのリストを提供し、ネットワーク、セキュリティ、およびその他のプロパティを設定します。起動テンプレートの詳細については、[Amazon EC2 User Guide] (Amazon EC2 ユーザーガイド) の[\[Launching an Instance from a Launch Template\]](#) (起動テンプレートからのインスタンスの起動) を参照してください。起動テンプレートはリージョン固有です。
- AWS IAM ロール IAM ロールは、AWS リソースへの制限付きアクセスを許可する一連のアクセス許可を定義します。別の AWS サービスなどの信頼されたエンティティは、ロールを引き受け、そのアクセス許可を継承できます。を使用する場合はAmazon GameLift ServersFleetIQ、AWS アカウントで Auto Scaling グループと EC2 インスタンスリソースの作成とアクセスAmazon GameLift ServersFleetIQを許可する マネージドポリシーを IAM ロールに提供する必要があります。IAM ロールはリージョン固有ではありません。

Amazon GameLift ServersFleetIQ は、以下のリソースを直接管理し、それらに対する直接的な権限を持ちます。

- Amazon GameLift Servers ゲームサーバーグループ。ゲームサーバーグループには、が対応する Auto Scaling グループAmazon GameLift ServersFleetIQと連携して低コストのゲームホスティングを提供する方法を定義する設定が含まれています。ゲームサーバーグループはリージョン固有です。リージョンでゲームサーバーグループを作成すると、同じリージョンの AWS アカウントで新しい Auto Scaling グループが自動的に作成されます。ゲームサーバーグループは Auto Scaling グループにリンクされ、その設定の一部を管理および変更するためのアクセス権を (IAM ロールを引き受けることによって) 付与されます。ゲームサーバーグループは寿命の長いリソースです。開

発者がこれらのグループを頻繁に作成することはありません。ゲームサーバーグループは、Auto Scaling グループのインスタンスでホストされ、に登録されているゲームサーバーの機能グループ リソースでもあります Amazon GameLift Servers FleetIQ。

- Amazon GameLift Servers ゲームサーバー。ゲームサーバーリソースは、ゲームサーバーグループに関連付けられたインスタンスで実行されている Amazon GameLift Servers FleetIQ ゲーム実行を表します。このリソースは、ゲームサーバーがに登録 Amazon GameLift Servers FleetIQ し、それが属するゲームサーバーグループを識別したときに作成されます。は、登録された各ゲームサーバーの使用率ステータスとクレームステータス Amazon GameLift Servers FleetIQ を追跡し、ゲームサーバーの可用性をモニタリングできるようにします。ゲームサーバーは、リージョン固有のゲームサーバーグループに関連付けられるという点で、リージョン固有です。ゲームは、新しいゲームサーバーをリクエストするときに、ゲームサーバーグループとリージョンを指定します。

これらのリソースは、 Amazon GameLift Servers FleetIQ リソースを通じて作成されます。これらは AWS アカウントに作成され、ユーザーが完全に制御できます。

- [Amazon EC2 Auto Scaling group] (Amazon EC2 Auto Scaling グループ) Auto Scaling グループは、EC2 インスタンスのコレクションを起動および管理し、グループの容量を自動的にスケールします。では Amazon GameLift Servers FleetIQ、ゲームサーバーグループと Auto Scaling グループの間に one-to-one の関係があります。Auto Scaling グループのすべての設定を更新できますが、は、そのロジックの一部として特定の設定 Amazon GameLift Servers FleetIQ を定期的に上書きして更新し、ゲームホスティングの実行可能性のためにスポットインスタンスのバランスを取ります。詳細については、[Amazon EC2 Auto Scaling User Guide] (Amazon EC2 Auto Scaling ユーザーガイド) の [\[AutoScalingGroup\]](#) (オートスケーリンググループ) を参照してください。Auto Scaling グループはリージョン固有で、ゲームサーバーグループと同じリージョンに作成されます。
- [Amazon EC2 Instance] (Amazon EC2 インスタンス) インスタンスとは、クラウドの仮想サーバーです。インスタンスタイプには、コンピューティング、メモリ、ディスク、およびネットワークリソースを指定する特定のハードウェア設定があります。通常、インスタンスは AMI を使用して Auto Scaling グループによって起動されます。インスタンスは、可用性に応じて、スポットまたはオンデマンドにすることができます。では Amazon GameLift Servers FleetIQ、インスタンスは 1 つ以上のゲームサーバープロセスを実行し、それぞれが複数のゲームセッションをホストできます。インスタンスは、リージョン固有の Auto Scaling グループに関連付けられるという点で、リージョン固有です。

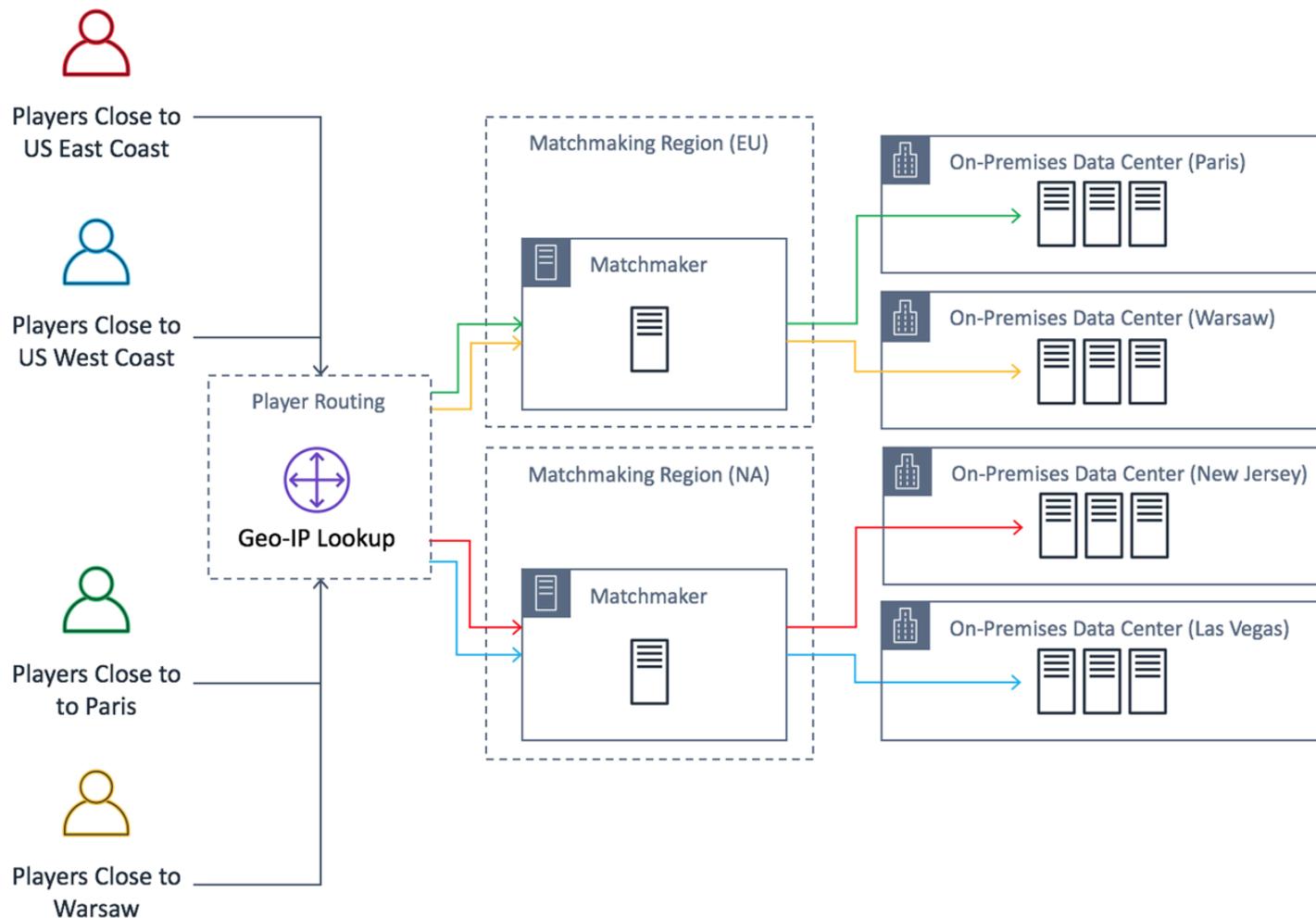
を使用したゲームアーキテクチャ Amazon GameLift ServersFleetIQ

オンプレミスホスティングの補完

Amazon GameLift ServersFleetIQ は、プレイヤーのジオ IP ルーティング、マッチメイキング、ロビーサービスなど、既存のゲームバックエンドを再利用するように設計されています。次の例は、Amazon GameLift ServersFleetIQが既存のオンプレミスデプロイにどのように適合するかを示しています。

Example

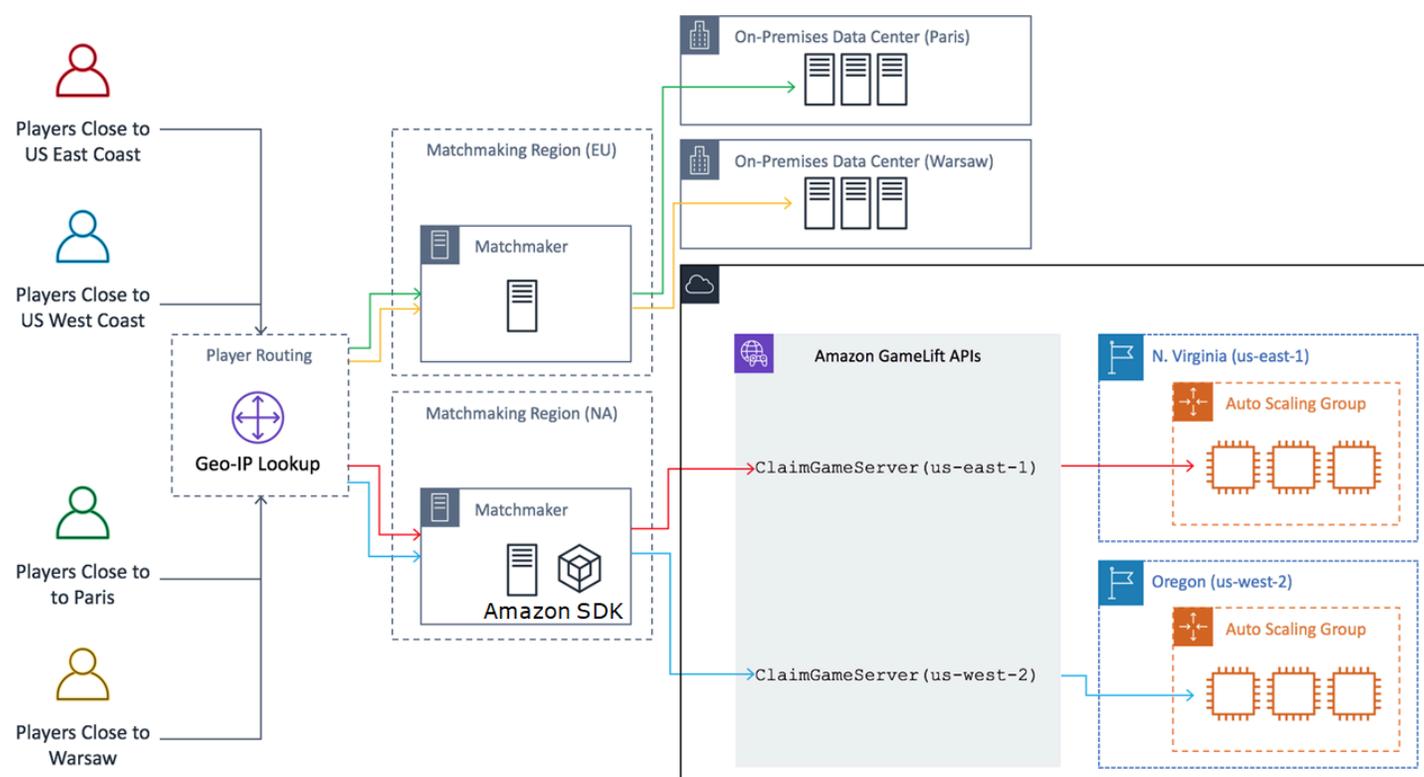
この例では、北米と欧州のプレイヤーをホストするための 4 つの専用データセンターで、ゲームホスティングが最初に処理されます。おおよその物理的な位置に応じて、プレイヤーは 2 つのリージョンマッチメーカーのいずれかにルーティングされます。マッチメーカーは、プレイヤーをスキルとレイテンシーに基づいてグループ化し、ラグを最小限に抑えるために近くのゲームサーバーに配置します。



ゲーム開発者は、北米のゲームサーバーを が提供するサーバーに置き換えたいと考えています Amazon GameLift ServersFleetIQ。まず、ゲームサーバーをマイナーアップデートして で使用できるようにしAmazon GameLift ServersFleetIQ、Amazon マシンイメージ (AMI) を作成します。このイメージは、ゲーム用にデプロイされるすべての EC2 インスタンスにインストールされます。イメージには、ゲームサーバー、依存関係、およびプレイヤーのゲームセッションを実行するために必要なあらゆるものが含まれています。

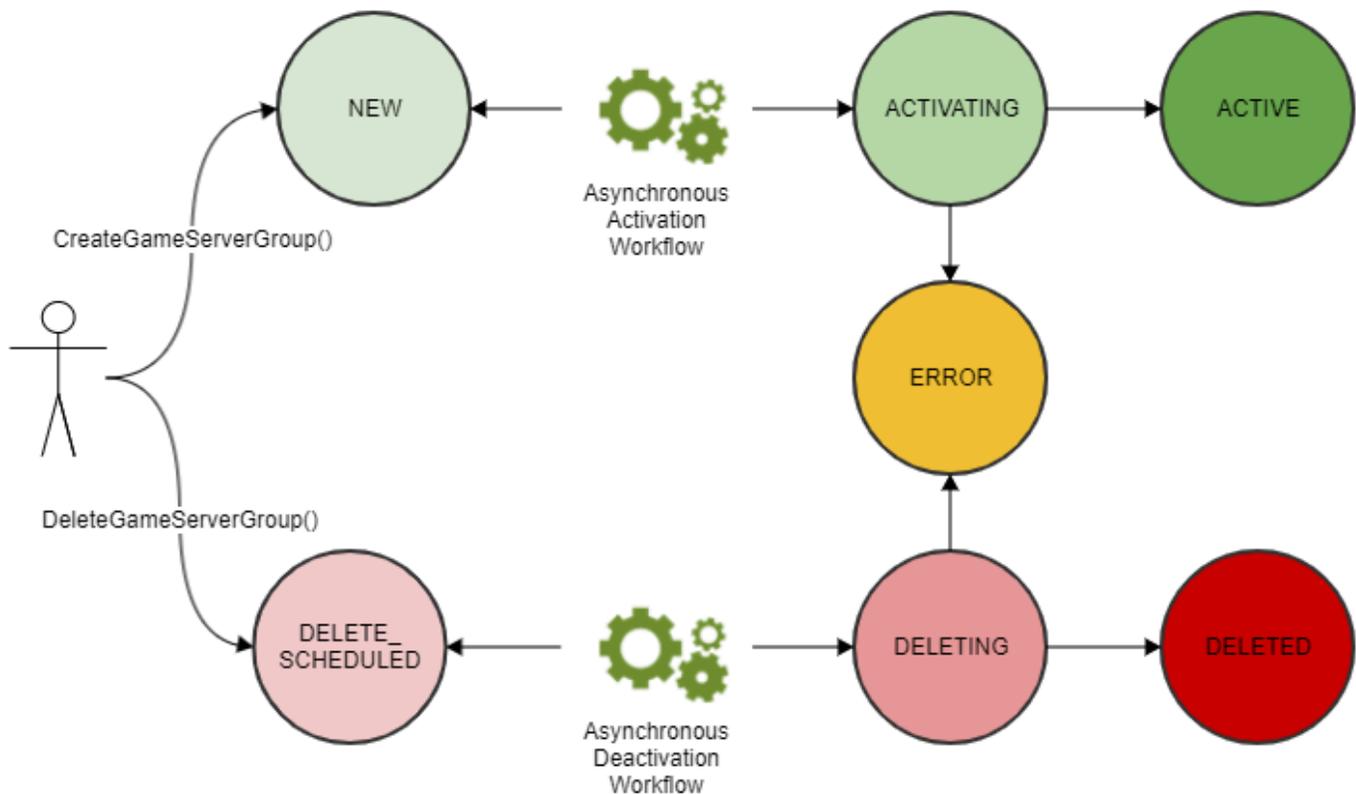
AMI の準備ができれば、デベロッパーは AWS 2 つの Amazon GameLift ServersFleetIQ ゲームサーバーグループを作成します。1 つは北米リージョン us-east-1 (と) ごとに作成します us-west-2)。デベロッパーは、起動テンプレート (AMI を提供)、必要なインスタンスタイプのリスト、およびグループのその他の設定を渡します。目的のインスタンスタイプのリストは、ゲームホスティングに有効なスポットインスタンスをチェックするとき使用する Amazon GameLift ServersFleetIQ タイプを に指示します。

最後に、デベロッパーは AWS SDK Amazon GameLift ServersFleetIQを北米のマッチメーカーに統合します。このマッチメーカーは、新しいプレイヤーグループがゲームセッションにサーバー容量を必要とするAmazon GameLift ServersFleetIQときに を呼び出します。は、利用可能なゲームサーバーを持つスポットインスタンスAmazon GameLift ServersFleetIQを見つけ、プレイヤーのために予約し、サーバー接続情報を提供します。プレイヤーはサーバーに接続して、ゲームを実行し、切断します。新しいゲームを開始するには、プレイヤーはマッチメイキングに再入力し、利用可能な別のゲームサーバーを見つけるAmazon GameLift ServersFleetIQように求められます。新しいゲームリクエストごとにトリガーAmazon GameLift ServersFleetIQされ、中断の可能性が低いゲームサーバーを検索して選択します。その結果、Amazon GameLift ServersFleetIQは、スポットインスタンスの可用性が時間の経過とともに変動しても、ゲームホスティングに使用できないゲームサーバーからプレイヤーを常にリダイレクトします。



ゲーム サーバーグループの寿命

ゲーム サーバーグループは、プロビジョニングやステータスの更新を含む、次のライフサイクルを経過します。ゲームサーバーグループは、寿命の長いリソースと想定されます。



- ゲームサーバーグループを作成するには、Amazon GameLift Servers API `CreateGameServerGroup()` を呼び出し、EC2 起動テンプレートと構成設定を渡します。呼び出しに応じて、新しいゲームサーバーグループが作成され、ステータスが `NEW` に設定されます。
- Amazon GameLift ServersFleetIQ は非同期アクティベーションワークフローをアクティブ化し、ゲームサーバーグループのステータスを `ACTIVATING` に移行します。このワークフローでは、Amazon EC2 Auto Scaling グループや、AMI が指定された EC2 インスタンスなど、基盤となるリソースの作成を開始します。
- 何らかの原因でプロビジョニングが失敗すると、ゲームサーバーグループのステータスは `ERROR` になります。エラーの原因のデバッグに役立つ追加のエラー情報を取得するには、エラー状態のゲームサーバーグループで `DescribeGameServerGroup()` を呼び出します。
- プロビジョニングが成功すると、ゲームサーバーグループのステータスは `ACTIVE` に移行します。この時点で、登録されているゲームサーバーでインスタンスが起動されます Amazon GameLift ServersFleetIQ。グループのインスタンスタイプは、ゲームホスティングの実行可能性について定期的に評価され、必要に応じて調整されます。Amazon GameLift ServersFleetIQ は、グループ内のアクティブなゲームサーバーのステータスを追跡し、ゲームサーバーのリクエストに応答します。

- ゲームサーバーグループを削除するには、グループ識別子を使用して `DeleteGameServerGroup()` を呼び出します。このアクションにより、ゲームサーバーグループのステータスは `DELETE_SCHEDULED` になります。削除対象となるのは、`ACTIVE` 状態または `ERROR` 状態のゲームサーバーグループのみです。
- Amazon GameLift Servers FleetIQ は、`DELETE_SCHEDULED` ステータスに応じて非同期非アクティブ化ワークフローをアクティブ化し、ゲームサーバーグループのステータスを `DELETING` に移行します。ゲームサーバーグループのみを削除するか、ゲームサーバーグループとリンクされた Auto Scaling グループの両方を削除するかを選択できます。
- 何らかの原因で非アクティブ化が失敗すると、ゲームサーバーグループのステータスは `ERROR` になります。エラーの原因のデバッグに役立つ追加のエラー情報を取得するには、エラー状態のゲームサーバーグループで `DescribeGameServerGroup()` を呼び出します。
- 非アクティブ化に成功すると、ゲームサーバーグループのステータスは `DELETED` に移行します。

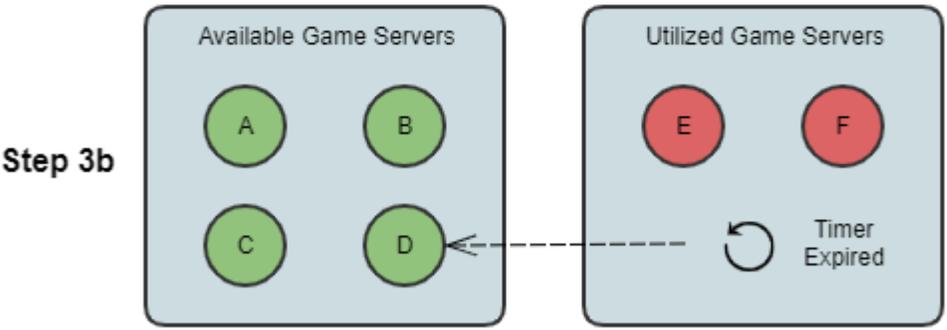
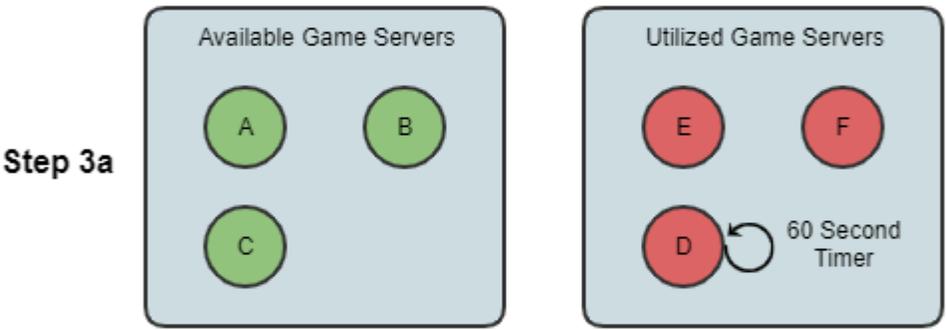
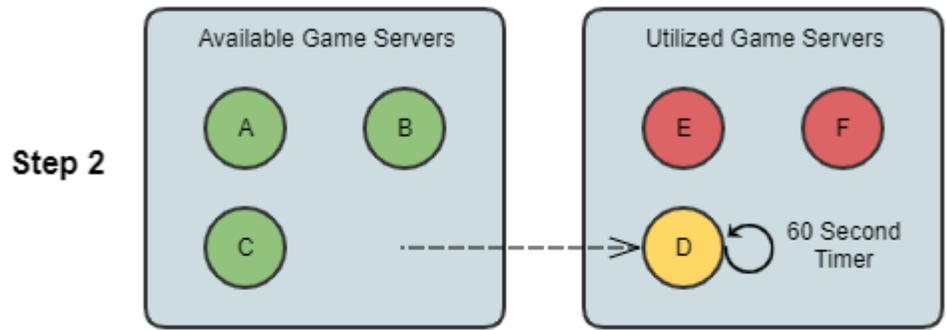
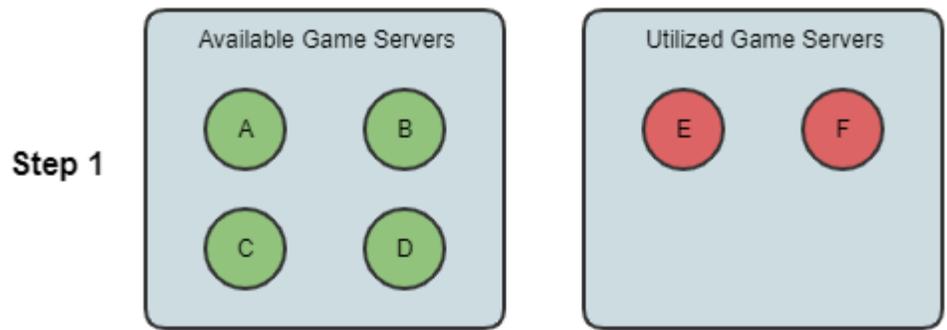
ゲームサーバーの寿命

では Amazon GameLift Servers FleetIQ、ゲームサーバーはプロビジョニングやステータスの更新など、次のライフサイクルを経ます。ゲームサーバーは寿命が短いリソースと想定されます。ベストプラクティスとして、ゲームサーバーは、ゲームセッションの終了後に他のゲームセッションに再利用せずにメンバーを解除する必要があります。これにより、使用可能なゲームサーバーは、ゲームホスティングのために最低コストで利用できるリソースで常に実行されることを確実にします。

- ゲームサーバーリソースは、Amazon GameLift Servers FleetIQ リンクされた Auto Scaling グループのインスタンスで実行されているゲームサーバープロセスが Amazon GameLift Servers API `RegisterGameServer()` を呼び出して、プレイヤーとゲームプレイをホストする準備ができてい Amazon GameLift Servers FleetIQ することを通知するときに作成されます。ゲームサーバーには、現在の可用性を追跡するための 2 つのステータスがあります。
- 使用状況ステータスは、ゲームサーバーがゲームプレイを現在サポートしているかどうかを追跡します。このステータスは最初に `AVAILABLE` に設定され、新しいゲームプレイを受け入れる準備ができていることを示します。ゲームサーバーがゲームプレイに使用されると、このステータスは `UTILIZED` に設定されます。
- クレームステータスは、ゲームサーバーが現在のゲームプレイにクレームされているかどうかを追跡します。 `CLAIMED` ステータスのゲームサーバーは、ゲームクライアント (またはマッチメーカーなどのゲームサービス) によって一時的に予約されていることを示します。このステータスにより Amazon GameLift Servers FleetIQ、 は同じゲームサーバーを複数のリクエストに提

供できなくなります。クレームステータスが空白になっているゲームサーバーは、クレームすることができます。

- 次の図は、ゲームサーバーの存続期間中に変わる使用状況ステータスとクレームステータスを示しています。



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

- [Step 1.](ステップ 1。)ゲームサーバーグループには、6 つのゲームサーバーが登録されています。使用状況ステータスは、4 つのゲームサーバーが AVAILABLE (A、B、C、および D)、2 つのゲームサーバーが UTILIZED (E および F) です。
- [Step 2.](ステップ 2。)ゲームクライアントまたはマッチメイキングシステムは Amazon GameLift Servers API `ClaimGameServer()` を呼び出して新しいゲームサーバーをリクエストします。このリクエスト Amazon GameLift Servers FleetIQ は、使用可能なゲームサーバー (D) を検索し、クレームステータスを 60 秒間 CLAIMED に設定するように求めます。は、ゲームサーバー (IP アドレスとポート) の接続情報と、その他のオプションのゲーム固有のデータを使用してリクエストに Amazon GameLift Servers FleetIQ 応答します。ゲームプレイはまだゲームサーバーで開始されていないため、その使用状況ステータスは AVAILABLE のままですが、別のリクエストによってクレームすることはできません。
- Step 3a. ゲームクライアントは、提供された接続情報を使用してゲームサーバーに接続し、ゲームプレイを開始できます。ゲームサーバー (D) は、60 秒以内にトリガーすることで、Amazon GameLift Servers API `UpdateGameServer()` を呼び出して使用状況ステータスを UTILIZED に変更する必要があります。
- Step 3b. ゲームサーバーの使用状況ステータスが 60 秒以内に更新されない場合、クレームタイマーの有効期限が切れて、クレームステータスが空白にリセットされます。ゲームサーバー (D) は、クレームされていない、使用可能なゲームサーバーのプールに戻されます。
- ゲームサーバーのゲームプレイが完了して、プレイヤーが切断されると、ゲームサーバーリソースは削除されます。シャットダウンする前に、ゲームサーバープロセスは Amazon GameLift Servers API `DeregisterGameServer()` を呼び出して、ゲームサーバーグループのゲームサーバーのプールからの Amazon GameLift Servers FleetIQ 逸脱を通知します。

スポット再調整プロセス

Amazon GameLift Servers FleetIQ は、スポットインスタンスを持つ Auto Scaling グループのインスタンスを定期的にバランスします。このプロセスは、ON_DEMAND_ONLY バランシング戦略を使用する、またはアクティブなインスタンスを持たないゲームサーバーグループではアクティブではありません。

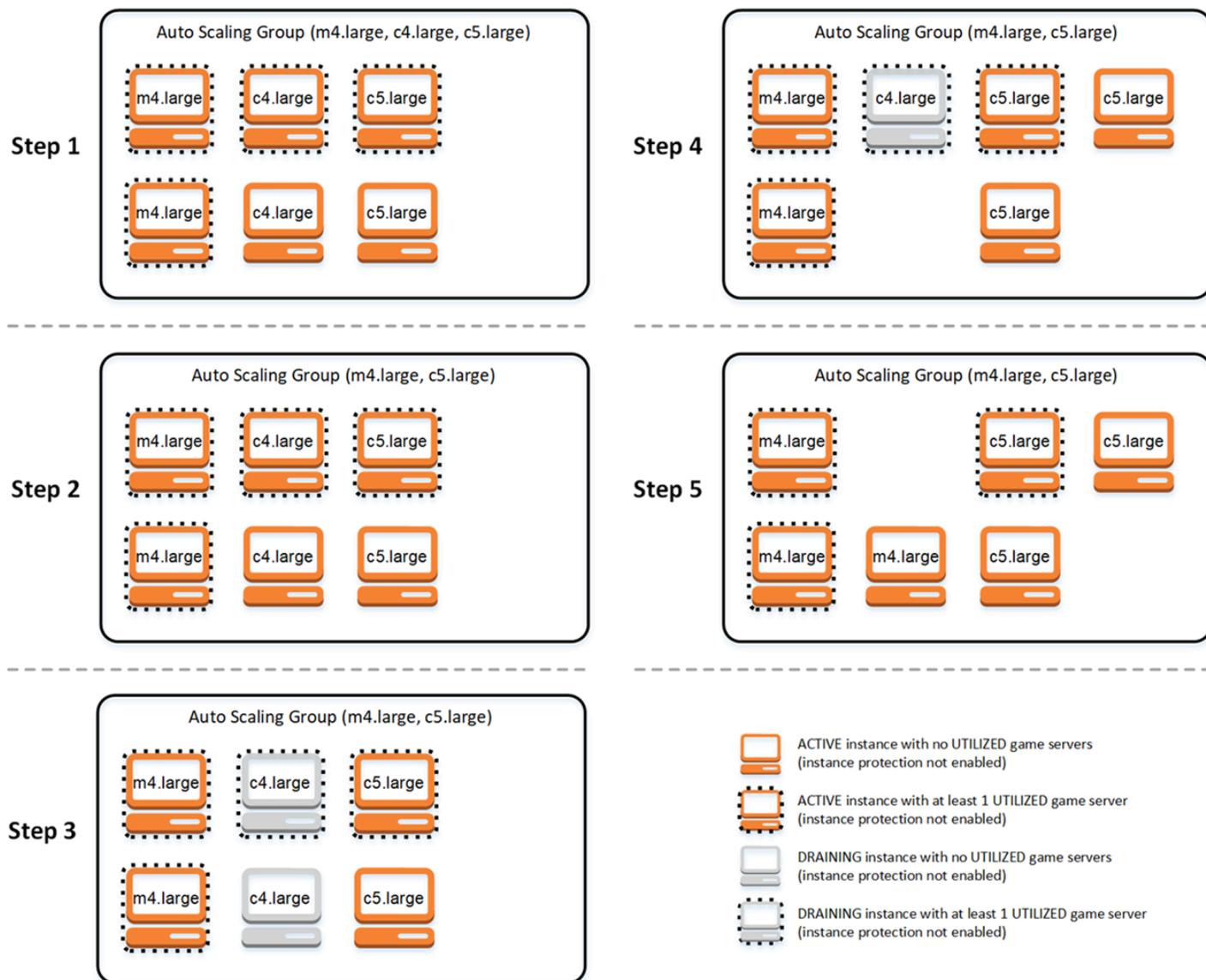
スポット再調整には、次の 2 つのキー目標があります。

- その時点でゲームをホスティングするための有効性が高いスポットインスタンスタイプのみを使用して、グループを常に更新することです。
- 予想外のゲームサーバー中断の影響を軽減するために、複数の有効なインスタンスタイプを使用することです (可能な場合)。

Amazon GameLift Servers FleetIQ は、グループのインスタンスタイプを評価し、ゲームサーバーの中断につながる可能性が高いインスタンスを削除することでバランスを取ります。再調整中にアクティブなゲームプレイのインスタンスを終了させないように、ベストプラクティスは、本番環境のゲームサーバーグループに対してゲームサーバーの保護を有効にすることです。

Example

次の例は、スポットの再調整が Auto Scaling グループのインスタンスに与える影響を示しています。



- Step 1. ゲームサーバーグループを介して、リンクされた Auto Scaling グループは、ゲームサーバーの保護が有効になっている m4.ラージ、c4.ラージ、c5.ラージ の各タイプのインスタンスを起動するように設定されます。Auto Scaling グループは、各タイプの 2 つのスポットインスタンスで

構成されるバランスの取れたコレクションを起動しました。4つのインスタンスには UTILIZED 状態 (破線の境界線で表示) のゲームサーバーが少なくとも 1 つあり、2 つのインスタンスはゲームプレイを現在サポートしていません。

- ステップ 2. は、3 つすべてのインスタンスタイプの現在のゲームホスティングの実行可能性 Amazon GameLift ServersFleetIQ を評価します。この評価により、c4.large インスタンスタイプにゲームサーバーの中断の可能性がないと判断されます。は Auto Scaling グループ設定を Amazon GameLift ServersFleetIQ 直ちに更新してインスタンスタイプのリストから c4.large を一時的に削除し、追加の c4.large インスタンスが起動されないようにします。
- ステップ 3. は c4.large タイプの既存のインスタンス Amazon GameLift ServersFleetIQ を識別し、グループから削除するアクションを実行します。最初のステップとして、c4.ラージ インスタンスで実行されているすべてのゲームサーバーに [draining](ドレイン中)のフラグが付けられます。ドレイン中のインスタンスのゲームサーバーは、他のゲームサーバーが使用できない場合の最終手段としてのみクレームできます。また、ドレイン中のインスタンスがある Auto Scaling グループはトリガーされ、新しいインスタンスが起動されてドレイン中のインスタンスと置き換えられます。
- Step 4. 新しい有効なインスタンスがオンラインになると、Auto Scaling グループはドレイン中のインスタンスを終了します。この置換により、グループの必要な容量が維持されます。最初に終了されるインスタンスは、使用中のゲームサーバーがなく、ゲームサーバーの保護がオフになっている c4.large インスタンスです。これは、新しい c5.large インスタンスに置き換えられます。
- Step 5. ゲームサーバーの保護が有効になっているドレイン中のインスタンスは、ゲームサーバーがゲームプレイをサポートしている間は引き続き実行されます。ゲームプレイが終了すると、残存していた c4.large インスタンスは、新しい m4.large インスタンスの起動時に置き換えられます。

このプロセスの結果として、Auto Scaling グループは希望する容量を維持し、グループは 3 つのインスタンスタイプの使用から 2 つのインスタンスタイプへのバランスを取ります。Amazon GameLift ServersFleetIQ はゲームホスティングの実行可能性について、インスタンスタイプの元のリストを引き続き評価します。c4.ラージ が再び有効なインスタンスタイプと見なされると、Auto Scaling グループが 3 つの インスタンスタイプのすべてを含むよう更新されます。時間が経つにつれて、グループのバランスが自然に取れてきます。

Amazon GameLift ServersFleetIQ ベストプラクティス

Amazon GameLift ServersFleetIQ は、ゲームホスティング用の Amazon EC2 リソースの管理に役立つ低レベルのロジックレイヤーです。特に、は、ゲームセッションが中断される可能性を最小限に抑えることで、ゲームホスティングに有効なスポットインスタンスの使用 Amazon GameLift ServersFleetIQ を最適化します。また、基本的なゲームホスティング機能により、利用可能なゲーム

サーバーを追跡し、できるだけ低コストで利用できるゲームサーバーにゲームプレイをルーティングします。

Amazon GameLift Servers FleetIQ は、 をスタンドアロン機能として使用しても、フルマネージド Amazon GameLift Servers ソリューションで提供される高度な機能を提供しません。また、 はホスティングコストを最小限に抑える FleetIQ ためにも使用します。マッチメイキング、レイテンシーベースのプレイヤールーティング、ゲームセッションとプレイヤーセッションの管理、バージョニングなどの機能が必要な場合は、 Amazon GameLift Servers ソリューションをご覧ください。

ここでは、 を最大限に活用するためのベストプラクティスをいくつか紹介します Amazon GameLift Servers FleetIQ。

- セッションベースのゲーム Amazon GameLift Servers FleetIQ に を使用します。 は、ゲームセッションの中断の可能性が低いインスタンスにプレイヤーを常に誘導している場合に最適です。 Amazon GameLift Servers FleetIQ 存続期間の長いセッションを維持すると、 Amazon GameLift Servers FleetIQ バランシングプロセスが妨げられ、ゲームセッションが中断される可能性が高くなります。理想的なワークフローは、プレイヤーがマッチメイキング (またはサーバー選択) からゲームプレイに移行することです。ゲームが終了すると、プレイヤーはマッチメイキングに戻り、新しいインスタンスの別のゲームサーバーにルーティングされます。セッションが 2 時間未満のゲーム Amazon GameLift Servers FleetIQ には、 を使用することをお勧めします。
- さまざまなインスタンスタイプを提供し、選択できるようにします。ゲームサーバーグループを設定する際に、使用するインスタンスタイプのリストを指定します。含めるインスタンスタイプが多いほど、ゲームホスティングの実行可能性の高いスポットインスタンスを柔軟に使用する Amazon GameLift Servers FleetIQ 必要があります。たとえば、同じインスタンスファミリー (c5.large、c5.xlarge、c5.2xlarge、c5.4xlarge) 内の複数のサイズをリストすることができます。インスタンスを大きくすると、各インスタンスでより多くのゲームサーバーを実行できるため、コストが削減される可能性があります。インスタンスを小さくすると、自動スケーリングがプレイヤーの需要の変化に迅速に対応できます。必要なインスタンスタイプのリストは優先順位付けされていないことに注意してください。Auto Scaling グループは、実行可能なインスタンスタイプのバランスを使用して、グループの復元力を維持します。
- すべてのインスタンスタイプでゲームをテストします。ゲームサーバーグループに設定したすべてのインスタンスタイプで、ゲームサーバーが正しく動作していることを確認します。
- インスタンス容量の重み付けを使用します。一連のインスタンスサイズ (c5.2xlarge、c5.4xlarge、c5.12xlarge など) を使用するようにゲームサーバーグループを構成する場合は、各インスタンスタイプの容量加重情報を含めます。詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[Amazon EC2 Auto Scaling インスタンスの重み付け](#)」を参照してください。

- を使用してゲームセッションを配置しますAmazon GameLift ServersFleetIQ。プレイヤーのグループをゲームサーバーに配置する場合は、Amazon GameLift Servers API を使用しますClaimGameServer()。は、ゲームセッションが中断される可能性の高いインスタンスにプレイヤーを配置するのAmazon GameLift ServersFleetIQを防ぎます。
- ゲームサーバーのステータスを に報告しますAmazon GameLift ServersFleetIQ。Amazon GameLift Servers API UpdateGameServer() を使用して、サーバーの動作状態と使用状況のステータスを定期的に報告します。正確なゲームサーバーのステータスを維持することで、ゲームプレイをより効率的にAmazon GameLift ServersFleetIQ配置できます。また、スポットのバルランシングアクティビティ中に、進行中のゲームプレイのインスタンスの終了を回避できます。
- Auto Scaling ポリシーを設定します。ターゲット追跡スケールリングポリシーを作成し、プレイヤーの使用状況と需要予測に基づいてホスティング容量を維持できます。Amazon GameLift ServersFleetIQ メトリクスPercentUtilizedGameServersは、現在使用されているホスティング容量の尺度です。ほとんどのゲームでは、未使用のゲームサーバーのバッファを維持し、新しいプレイヤーがすぐにゲームに参加できるようにする必要があります。特定のバッファサイズを維持するスケールリングポリシーを作成することで、プレイヤーの需要の変動に応じてインスタンスを追加または削除できます。詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[ターゲット追跡スケールリングポリシー](#)」を参照してください。
- 開発環境と本番環境には、異なる AWS アカウントを使用します。複数のアカウントを開発設定と実稼働設定で使い分けることで、不適切な設定がライブプレイヤーにもたらすリスクを軽減できます。
- 実稼働環境でゲームサーバーグループのゲームセッション保護を有効にします。プレイヤーを保護するには、ゲームセッションの保護をオンにし、アクティブなゲームセッションがスケールリングやバルランシングアクティビティによって早期に終了しないようにします。
- と統合する前にEC2 でゲームをテストしますAmazon GameLift ServersFleetIQ。EC2 でゲームを起動して実行し、最初に設定を微調整することをお勧めします。その後、同じ起動テンプレートとAMI を使用してゲームサーバーグループを作成できます。

Kubernetes を使用している場合は、まず Kubernetes クラスターに標準 EC2 インスタンスを追加してから、Kubernetes クラスターのワーカーノード用に作成した起動テンプレートを使用してゲームサーバーグループを作成することをお勧めします。EKS を使用している場合は、EKS クラスターとゲームサーバーグループを個別に作成します。ゲームサーバーグループでは、EKS 統合に使用する適切なユーザーデータおよび起動テンプレート設定とともに、EBS に最適化したAMIを使用します。EKS ワーカーノードとEKS最適化AMIの詳細については、[Amazon EKS 最適化Linux AMI](#) ガイドを参照してください。

- ゲームサーバーグループのバランス戦略 **ON_DEMAND_ONLY** を使用し信頼できるゲームサーバーの可用性を確保します。このバルランシング戦略が有効な場合、スポットインスタンスは使用されませ

ん。この機能は、新機能の発表やその他の特別なイベントなど、最も必要なときにサーバーの可用性を確保するのに便利なツールです。必要に応じて、ゲームサーバーグループをスポットからオンデマンド戦略に切り替えることができます。

また、以下の AWS ベストプラクティスも確認してください。

- [Amazon EC2 のベストプラクティス](#)
- [Amazon EC2 Auto Scaling のベストプラクティス](#)

Amazon GameLift Servers features FleetIQ

- スポットバランシングの最適化。は、インスタンスタイプ Amazon GameLift Servers FleetIQ を定期的に評価し、ゲームセッションの中断の可能性が高いために実行可能と見なされないスポットインスタンスを置き換えます。EC2 Auto Scaling グループが古いインスタンスを廃止して新しいインスタンスを開始すると、グループはゲームホスティングで現在実行可能なインスタンスタイプで継続的に更新されます。
- 最適なプレイヤールーティング。Amazon GameLift Servers FleetIQ APIs、新しいゲームセッションを最も回復力の高いスポットインスタンスに誘導します。この場合、中断される可能性は最も低くなります。また、ゲームセッションは少数のインスタンスにまとめられるため、不要なリソースをスケールダウンする EC2 Auto Scaling グループの能力が向上し、コスト削減につながります。
- プレイヤーの使用状況に基づく自動スケーリング。Amazon GameLift Servers FleetIQ は、ゲームサーバー使用率データを Amazon CloudWatch メトリクスとして出力します。これらのメトリクスを使用することで、実際のプレイヤーの需要に合わせて利用可能なホスティングリソースを自動的に測定し、ホスティングコストを削減します。
- Amazon EC2 インスタンスの直接管理。AWS アカウントの EC2 インスタンスと Auto Scaling グループの完全な制御を維持します。つまり、インスタンス起動テンプレートを設定し、EC2 Auto Scaling グループ設定を維持し、他の AWS サービスと統合できます。スポットバランシングアクティビティの一環として、Amazon GameLift Servers FleetIQ は一部の EC2 Auto Scaling グループプロパティを定期的に更新します。必要に応じて、これらの設定を一時的に上書きしたり、Amazon GameLift Servers FleetIQ アクティビティを停止したりできます。
- 複数のゲームサーバー実行可能形式のサポート。は、Windows、Linux、コンテナ、Kubernetes など、Amazon EC2 で現在実行されているすべての形式 Amazon GameLift Servers FleetIQ をサポートしています。サポートされているオペレーティングシステムとランタイムの一覧については、[Amazon EC2 FAQs](#) (Amazon EC2 のよくある質問) を参照してください。

- 複数のタイプのホスティングリソース。を使用するとAmazon GameLift ServersFleetIQ、ゲームサーバーホスティング用の幅広いインスタンスタイプにアクセスできます。(可用性はAWSリージョンによって異なります)。これにより、ゲームサーバーをCPU、メモリ、ストレージ、およびネットワーク容量の適切な組み合わせとペアリングして、プレイヤーに最適なゲーム体験を提供できます。
- ワールドワイドリーチ。Amazon GameLift ServersFleetIQは、中国を含む15のリージョンで利用できます。この規模のリーチにより、プレイヤーの場所を問わず、最小限のラグでゲームサーバーにアクセスできます。リージョンの完全なリストについては、の[Amazon GameLift Servers「エンドポイントとクォータ」](#)を参照してくださいAWS全般のリファレンス。

Amazon GameLift ServersFleetIQ の料金

Amazon GameLift Servers は、インスタンスの使用期間および転送されるデータ量別の帯域幅に対して課金します。Amazon GameLift Servers の課金および特定の料金の詳細な一覧については、「[Amazon GameLift Servers の料金](#)」を参照してください。

でのゲームまたはマッチメイキングのホスティングコストの計算についてはAmazon GameLift Servers、「[の使用方法を説明するAmazon GameLift Servers料金見積りの生成](#)」を参照してください[AWS 料金見積りツール](#)。

Amazon GameLift ServersFleetIQ のセットアップ

このセクションのトピックは、Amazon Amazon GameLift ServersFleetIQサービスで使用するアカウント AWS を設定する方法など、タスクの設定に役立ちます。

トピック

- [Amazon GameLift ServersFleetIQ サポートされているソフトウェア](#)
- [の AWS アカウントをセットアップする Amazon GameLift ServersFleetIQ](#)

Amazon GameLift ServersFleetIQ サポートされているソフトウェア

Amazon GameLift ServersFleetIQ は、Amazon EC2 でホストするための 64 ビット、マルチプレイヤーゲームサーバー、クライアント、ゲームサービスをデプロイするために使用されます。このソリューションは、次の環境をサポートします。

ゲームサーバー用オペレーティングシステム

は、EC2 でサポートされている任意のオペレーティングシステムで実行されるゲームサーバーAmazon GameLift ServersFleetIQで使用できます。これには、Amazon Linux、Ubuntu、Windows Server、Red Hat Enterprise Linux、SUSE Linux Enterprise Server、Fedora、Debian、CentOS、Gentoo Linux、Oracle Linux、FreeBSD が含まれます。現在の EC2 の特徴と Support については、[\[Amazon EC2 features\]](#) (Amazon EC2 の特徴) を参照してください。

コンテナの使用

ゲームサーバーがコンテナを使用している場合、は Kubernetes、Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (EKS) との統合 Amazon GameLift ServersFleetIQ をサポートします。詳細については、「[AWS のコンテナ](#)」を参照してください。

ゲーム開発環境

ゲームクライアントとサーバーは、Amazon GameLift ServersFleetIQ サービスと通信するために何らかの統合が必要です。ゲームは AWS SDK に API コールを行います。[AWS SDK をダウンロードするか、Amazon GameLift ServersAPI リファレンスドキュメントを参照してください。](#)

をサポートする AWS SDK Amazon GameLift Serversは、次の言語で使用できます。開発環境のサポートの詳細については、各言語のドキュメントを参照してください。

- C++ ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- Java ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- .NET ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- Go ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- Python ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- Ruby ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- PHP ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- JavaScript/Node.js ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))

の AWS アカウントをセットアップする Amazon GameLift ServersFleetIQ

Amazon EC2、Auto Scaling、およびその他の AWS サービスAmazon GameLift ServersFleetIQで を使用するには、必要なアクセス許可を持つ AWS アカウント を設定する必要があります。以下のタスクを実行します。

- 使用するアカウントがまだない場合はAmazon GameLift ServersFleetIQ、新しい AWS アカウントを作成します。「[を作成する AWS アカウント](#)」を参照してください。
- ユーザーとユーザーグループにAmazon GameLift ServersFleetIQ固有のアクセス許可を設定します。「[のユーザーアクセス許可を管理する Amazon GameLift ServersFleetIQ](#)」を参照してください。
- Amazon GameLift Servers と Amazon EC2 リソースのやり取りを許可する IAM ロールを作成します。「[サービス間でやり取りするための IAM ロールの作成](#)」を参照してください。

を作成する AWS アカウント

AWS アカウント で使用する を作成してセットアップしますAmazon GameLift ServersFleetIQ。AWS アカウントの作成には料金はかかりません。

トピック

- [にサインアップする AWS アカウント](#)

• [管理アクセスを持つユーザーを作成する](#)

にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. <https://portal.aws.amazon.com/billing/signup> を開きます。
2. オンラインの手順に従います。

サインアップ手順の一環として、電話またはテキストメッセージを受け取り、電話キーパッドで検証コードを入力します。

にサインアップすると AWS アカウント、AWS アカウントのルートユーザー が作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティベストプラクティスとして、ユーザーに管理アクセス権を割り当て、[ルートユーザーアクセスが必要なタスク](#)の実行にはルートユーザーのみを使用するようにしてください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。<https://aws.amazon.com/> の [マイアカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

管理アクセスを持つユーザーを作成する

にサインアップしたら AWS アカウント、日常的なタスクにルートユーザーを使用しないように AWS アカウントのルートユーザー、 を保護し AWS IAM Identity Center、 を有効にして管理ユーザーを作成します。

を保護する AWS アカウントのルートユーザー

1. ルートユーザーを選択し、AWS アカウント E メールアドレスを入力して、アカウント所有者 [AWS Management Console](#) として にサインインします。次のページでパスワードを入力します。

ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの [ルートユーザーとしてサインインする](#) を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

手順については、IAM [ユーザーガイドの AWS アカウント「ルートユーザー \(コンソール\) の仮想 MFA デバイス](#)を有効にする」を参照してください。

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

手順については、「AWS IAM Identity Center ユーザーガイド」の「[AWS IAM Identity Centerの有効化](#)」を参照してください。

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

を ID ソース IAM アイデンティティセンターディレクトリとして使用する方法のチュートリアルについては、「AWS IAM Identity Center ユーザーガイド」の「[デフォルトを使用してユーザーアクセスを設定する IAM アイデンティティセンターディレクトリ](#)」を参照してください。

管理アクセス権を持つユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

IAM Identity Center ユーザーを使用してサインインする方法については、AWS サインイン「[ユーザーガイド](#)」の AWS 「[アクセスポータルにサインイン](#)する」を参照してください。

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

手順については、「AWS IAM Identity Center ユーザーガイド」の「[権限設定を作成する](#)」を参照してください。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

手順については、「AWS IAM Identity Center ユーザーガイド」の「[グループの結合](#)」を参照してください。

のユーザーアクセス許可を管理する Amazon GameLift ServersFleetIQ

必要に応じて、追加のユーザーを作成するか、既存のユーザー Amazon GameLift ServersFleetIQ にアクセス許可を拡張します。Amazon GameLift ServersFleetIQ ゲームサーバーグループおよび関連する Amazon EC2 および Auto Scaling サービスを操作するユーザーには、これらのサービスにアクセスするためのアクセス許可が必要です。

ベストプラクティス ([IAM におけるセキュリティのベストプラクティス](#)) として、すべてのユーザーに最小特権のアクセス権限を適用します。個々のユーザーまたはユーザーグループにアクセス許可を設定し、サービス、アクション、またはリソース別にユーザーアクセスを制限できます。

以下の手順を使用して、AWS アカウントのユーザーの管理方法に基づいてユーザーアクセス許可を設定します。IAM ユーザーを使用する場合、ベストプラクティスとして、個々のユーザーではなく、常にロールまたはユーザーグループにアクセス許可を付与してください。

- [ユーザーのアクセス許可構文](#)
- [で使用する追加のアクセス許可構文 AWS CloudFormation](#)

アクセス権限を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ AWS IAM Identity Center :

アクセス許可セットを作成します。「AWS IAM Identity Center ユーザーガイド」の「[権限設定を作成する](#)」の手順に従ってください。

- IAM 内で、ID プロバイダーによって管理されているユーザー:

ID フェデレーションのロールを作成します。詳細については「IAM ユーザーガイド」の「[サードパーティー ID プロバイダー \(フェデレーション\) 用のロールを作成する](#)」を参照してください。

- IAM ユーザー:

- ユーザーが担当できるロールを作成します。手順については「IAM ユーザーガイド」の「[IAM ユーザーのロールの作成](#)」を参照してください。
- (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。詳細については「IAM ユーザーガイド」の「[ユーザー \(コンソール\) へのアクセス権限の追加](#)」を参照してください。

リファレンス : Amazon GameLift ServersFleetIQ_policy

以下は、参照用の Amazon GameLift ServersFleetIQ_policy の例です。

```
{
  "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": "*"
  }
]
```

の追加アクセス許可 AWS CloudFormation

AWS CloudFormationを使用してゲームホスティングリソースを管理する場合は、ポリシー構文にアクセス AWS CloudFormation 許可を追加します。

```
{
  "Action": [
    "autoscaling:DescribeLifecycleHooks",
    "autoscaling:DescribeNotificationConfigurations",
    "ec2:DescribeLaunchTemplateVersions"
  ]
  "Effect": "Allow",
  "Resource": "*"
}
```

ユーザーにプログラムによるアクセスをセットアップする

ユーザーが の AWS 外部で を操作する場合は、プログラムによるアクセスが必要です AWS Management Console。プログラムによるアクセスを許可する方法は、 がアクセスするユーザーのタイプによって異なります AWS。

ユーザーにプログラマチックアクセス権を付与するには、以下のいずれかのオプションを選択します。

プログラマチックアクセス権を必要とするユーザー	目的	方法
ワークフォースアイデンティティ (IAM アイデンティティセンターで管理されているユーザー)	一時的な認証情報を使用して AWS CLI、AWS SDKs、または AWS APIs。	使用するインターフェイスの指示に従ってください。 <ul style="list-style-type: none"> については AWS CLI、AWS Command Line Interface ユーザーガイドの 「を使用する AWS CLI ように AWS IAM Identity Centerを設定する」を参照してください。 AWS SDKs、ツール、API については、AWS APIs 「SDK およびツールリファレンスガイド」の 「IAM アイデンティティセンター認証」を参照してください。 AWS SDKs
IAM	一時的な認証情報を使用して AWS CLI、AWS SDKs、または AWS APIs。	「IAM ユーザーガイド 」の 「AWS リソースでの一時的な認証情報の使用」 の手順に従います。
IAM	(非推奨)	使用するインターフェイスの指示に従ってください。

プログラマチックアクセス権を必要とするユーザー	目的	方法
	<p>長期認証情報を使用して、AWS CLI、AWS SDKs、または AWS APIs。</p>	<ul style="list-style-type: none"> • については AWS CLI、「AWS Command Line Interface ユーザーガイド」の「IAM ユーザー認証情報を使用した認証」を参照してください。 • AWS SDKs 「SDK とツールリファレンスガイド」の「長期認証情報を使用した認証」を参照してください。AWS SDKs • API AWS APIs 「IAM ユーザーガイド」の「IAM ユーザーのアクセスキーの管理」を参照してください。

アクセスキーを使用する場合は、[AWS 「アクセスキーを管理するためのベストプラクティス」](#)を参照してください。

サービス間でやり取りするための IAM ロールの作成

Amazon GameLift ServersFleetIQ が Amazon EC2 インスタンスと Auto Scaling グループを操作するには、サービスが相互にやり取りできるようにする必要があります。これは、AWS アカウントに IAM ロールを作成し、制限されたアクセス許可のセットを割り当てます。各ロールでは、ロールを引き受けるサービスも指定します。

次のロールを設定します。

- [のロールを作成する Amazon GameLift ServersFleetIQ](#) で Amazon EC2 リソースを更新します。
- [Amazon EC2 のロールを作成する](#) 通信する リソース Amazon GameLift ServersFleetIQ。

のロールを作成する Amazon GameLift ServersFleetIQ

このロールによりAmazon GameLift ServersFleetIQ、はスポットバランシングおよび自動スケールリングアクティビティの一環として、Amazon EC2 インスタンス、Auto Scaling グループ、ライフサイクルフックにアクセスして変更することができます。

IAM コンソールまたは AWS CLI を使用して、のロールを作成しAmazon GameLift ServersFleetIQ、必要なアクセス許可を持つ管理ポリシーをアタッチします。IAM ロールと管理ポリシーの詳細については、「[AWS サービスおよび管理ポリシーのロールの作成](#)」を参照してください。 [AWS](#)

Console

以下のステップでは、AWS Management Consoleを使用して Amazon GameLift Servers の管理ポリシーでサービスロールを作成する方法について説明します。

1. [\[IAM console\]](#) (IAM コンソール) にログインして、[Roles: Create role] (ロール: ロールの作成) を選択します。
2. [Select type of trusted entity] (信頼されたエンティティのタイプの選択) で、[AWS のサービス] を選択します。
3. [Choose a use case] (ユースケースの選択) で、サービスのリストから[GameLift]を選択します。[ユースケースの選択] で、適切な Amazon GameLift Servers ユースケースが自動的に選択されます。続行するには、[Next: Permissions] (次へ: アクセス許可) を選択します。
4. [Attached permissions policies] (アタッチされたアクセス許可ポリシー) のリストには、[GameLiftGameServerGroupPolicy] (GameLiftゲームサービスグループポリシー) というポリシーが 1 つ含まれている必要があります。このポリシーが表示されない場合は、フィルターを確認するか、検索機能を使用してロールに追加します。ポリシーの構文を表示できますが (▶ アイコンを選択して展開します)、構文を変更することはできません。ロールが作成されたら、ロールを更新し、追加のポリシーをアタッチして、アクセス許可を追加または削除できます。

[Set permissions boundary] (アクセス許可の境界の設定) では、既定の設定 (アクセス許可の境界のないロールを作成) のデフォルトのままにします。これは詳細設定は必須ではありません。続行するには [Next: Tags] (次へ: タグ) を選択します。

5. [Add tags] (タグの追加) は、リソース管理のオプション設定です。たとえば、このロールにタグを追加して、プロジェクト固有のリソース使用状況をロール別に追跡できます。IAM ロールやその他の用途のタグ付けの詳細については、[Learn more] (詳細はこちら) リンクを参照してください。[Next: Review] (次へ: 確認) を選択して続行します。
6. [Review] (確認) ページで、必要に応じて次の変更を行います。

- ロール名を入力し、必要に応じて説明を更新します。
- 以下について確認します。
 - 信頼されたエンティティはAWS 「service: gamelift.amazonaws.com」に設定されます。この値は、ロールの作成後に更新する必要があります。
 - [Policies] (ポリシー) には、GameLiftGameServerGroupPolicy が含まれます。

タスクを完了するには、[ロールの作成] を選択します。

7. 新しいロールが作成されたら、ロールの信頼関係をマニュアルで更新する必要があります。[ロール] ページに移動し、新しいロール名を選択して概要ページを開きます。[信頼関係] タブを開き、[信頼関係の編集] を選択します。ポリシードキュメントで、Service プロパティを更新して、[autoscaling.amazonaws.com] を含めます。変更された Service プロパティは次のようになります。

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

変更を保存するには、[信頼ポリシーの更新] を選択します。

これでロールの準備が整いました。ロールの ARN 値をメモします。この値は、ロールの概要ページの上部に表示されます。この情報は、Amazon GameLift Servers FleetIQ ゲームサーバーグループを設定するときに必要なになります。

AWS CLI

以下の手順では、CLI AWS Amazon GameLift Servers を使用するための マネージドポリシーを使用してサービスロールを作成する方法について説明します。

1. 次の JSON 構文を使用して、信頼ポリシーファイル (例: FleetIQtrustpolicyGameLift.json) を作成します。

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

```
    "Principal": {
      "Service": [
        "gamelift.amazonaws.com",
        "autoscaling.amazonaws.com"
      ]
    },
    "Action": "sts:AssumeRole"
  }
]
```

2. [iam create-role](#) (IAM Create-ロール) で新しい IAM ロールを作成します、先ほど作成した信頼ポリシー JSON ファイルに関連付けます。

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
```

リクエストが成功すると、応答には新しく作成したロールのプロパティが含まれます。ARN 値を書き留めます。この情報は、Amazon GameLift Servers FleetIQ ゲームサーバーグループを設定するときに必要なになります。

3. [iam attach-role-policy](#) を使用して、管理アクセス許可ポリシー "GameLiftGameServerGroupPolicy" をアタッチします。

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

アクセス許可ポリシーがアタッチされていることを確認するには、新しいロール名で [iam list-attached-role-policies](#) を呼び出します。

これでロールの準備が整いました。IAM ロールが正しく設定されていることを確認するには、role-arn プロパティを新しいロールの ARN 値に設定して、[gamelift create-game-server-group](#)(gamelift クリエイト ゲーム サーバー グループ)を呼び出します。が ACTIVE 状

態GameServerGroupになると、Amazon GameLift ServersFleetIQが想定どおりにアカウントの Amazon EC2 および Auto Scaling リソースを変更できることを示します。

Amazon EC2 のロールを作成する

このロールにより、Amazon EC2 リソースが通信できるようになりますAmazon GameLift ServersFleetIQ。たとえば、Amazon EC2 インスタンスで実行されているゲームサーバーは、ヘルステータスをレポートできる必要があります。Amazon GameLift ServersFleetIQ ゲームサーバーグループを作成するときに、このロールを Amazon EC2 起動テンプレートの IAM インスタンスプロファイルに含めます。

CLI を使用して Amazon EC2 AWS のロールを作成し、必要なアクセス許可を持つカスタムポリシーをアタッチし、そのロールをインスタンスプロファイルにアタッチします。詳細については、「[AWS サービスのロールの作成](#)」を参照してください。

AWS CLI

これらのステップでは、を使用して Amazon EC2 のカスタムAmazon GameLift Serversアクセス許可を持つサービスロールを作成する方法について説明します AWS CLI。

1. 次の JSON 構文を使用して、信頼ポリシーファイル (例: FleetIQtrustpolicyEC2.json) を作成します。

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

2. [\[iam create-role\]](#) (IAM Create-ロール) で新しい IAM ロールを作成します、先ほど作成した信頼ポリシー JSON ファイルに関連付けます。

Windows:

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

Linux:

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

リクエストが成功すると、応答には新しく作成したロールのプロパティが含まれます。ARN 値を書き留めます。この情報は、Amazon EC2 起動テンプレートを設定する場合に必要になります。

3. 次の JSON 構文を使用して、アクセス許可ポリシーファイル (例: FleetIQpermissionsEC2.json) を作成します。

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

4. [iam put-role-policy](#) を使用して、先ほど作成したアクセス許可ポリシー JSON ファイルを新しいロールにアタッチします。

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
```

アクセス許可ポリシーがアタッチされていることを確認するには、新しいロール名で [iam list-role-policies](#) を呼び出します。

5. [\[iam create-instance-profile\]](#) (iam インスタンスプロファイル) でインスタンスプロファイルを作成します。新しいロールは Amazon EC2 で使用します。詳細については、「[インスタンスプロファイルの管理](#)」を参照してください。

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

リクエストが成功すると、応答には新しく作成されたインスタンスプロファイルのプロパティが含まれるようになります。

6. [iam add-role-to-instance-profile](#) を使用して、インスタンスプロファイルにロールをアタッチします。

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

これで、ロールおよびプロファイルを Amazon EC2 起動テンプレートで使用する準備ができました。

のゲームの準備 Amazon GameLift ServersFleetIQ

このセクションでは、を使用して Amazon EC2 Amazon GameLift ServersFleetIQ でゲームをホストする設計を実装する方法について説明します。マルチプレイヤーゲームを起動して実行するには、以下を実行する必要があります。

- 通信するゲームサーバーを適応させます Amazon GameLift ServersFleetIQ。
- FleetIQ ゲームサーバーグループを作成して、ゲームサーバーをデプロイします。
- ゲームクライアントサービスに機能を追加し、利用可能なゲームサーバーをリクエストします。

このセクションのトピックでは、この作業の実行方法の詳細を提供します。開始するには、詳細なステップバイステップガイドを提供する統合プランを参照してください。

トピック

- [Amazon GameLift ServersFleetIQ 統合ステップ](#)
- [Amazon GameLift ServersFleetIQ ゲームサーバーグループの管理](#)
- [ゲームサーバーAmazon GameLift ServersFleetIQに統合する](#)
- [ゲームクライアントAmazon GameLift ServersFleetIQに統合する](#)

Amazon GameLift ServersFleetIQ 統合ステップ

この統合プランでは、を使用して Amazon EC2 インスタンスでマルチプレイヤーゲームを起動および実行するための主要なステップの概要を説明します Amazon GameLift ServersFleetIQ。より多くのゲームホスティングプロセスを自動化する Amazon GameLift Servers マネージドホスティングサービスをお探しの場合は、「[Amazon GameLift Serversデベロッパーガイド](#)」を参照してください。

の使用を開始するには Amazon GameLift ServersFleetIQ、オンプレミス環境または Amazon EC2 環境で実行されるゲームサーバーが必要です。ゲームサーバーは、1 つまたは複数のゲームセッションを管理したり、子プロセスを生成したり、コンテナ内で実行したりする単一のプロセスにすることができます。

1. [AWS アカウント](#) を取得し、Amazon GameLift ServersFleetIQ アクセス可能なユーザーを設定します。

新しいアカウントを作成する AWS アカウント か、使用する既存のアカウントを選択します Amazon GameLift ServersFleetIQ。ゲームで使用される Amazon EC2、Auto Scaling、およびその

他の AWS リソースを管理するアクセス許可を持つユーザーを設定します。詳細な手順については、「[の AWS アカウントをセットアップする Amazon GameLift ServersFleetIQ](#)」を参照してください。

2. IAM ロールを作成します。

Amazon GameLift ServersFleetIQ、Amazon EC2、および Auto Scaling リソースが相互に通信できるようにするロールを作成します。詳細については、「[サービス間でやり取りするための IAM ロールの作成](#)」を参照してください。

3. Amazon GameLift ServersFleetIQ機能を使用して AWS SDK と AWS CLI を取得します。

- [AWS SDK の最新バージョンをダウンロードする。](#)
- [Amazon GameLift Servers API リファレンスドキュメントを確認します。](#)

4. で使用するゲームサーバーを準備しますAmazon GameLift ServersFleetIQ。

AWS SDK をゲームサーバープロジェクトに追加し、ゲームサーバーの現在のステータスと使用状況でAmazon GameLift ServersFleetIQ更新するコードを追加します。追加のガイダンスと例[the section called “ゲームサーバーの統合”](#)については、「」を参照してください。Amazon GameLift ServersFleetIQはこの情報を使用して、有効で占有されていないゲームサーバーのリストをマッチングシステムに提供し、バランシング中に現在プレイヤーをホストしているインスタンスを終了しないようにします。

5. ゲームサーバーで Amazon EC2 Amazon マシンイメージ (AMI) を作成する。

ゲームサーバーソフトウェアと、その他のランタイムアセットまたは設定を使用して AMI を作成します。ヘルプについては、[Amazon Machine Images (AMI)] (Amazon EC2 ユーザーガイド) の「[Amazon マシンイメージ \(AMI\)](#)」を参照してください。

6. [Create an Amazon EC2 launch template] (Amazon EC2 起動テンプレートを作成する

カスタム AMI を使用し、ホスティングリソースのネットワークとセキュリティ設定を定義する Amazon EC2 起動テンプレートを構築します。起動テンプレートは、ゲームサーバーとの通信を許可するアクセス許可を持つ、作成したインスタンスプロファイルを参照する必要があります (ステップ 2 を参照) Amazon GameLift ServersFleetIQ。インスタンスタイプは後で含めるので、起動テンプレートにインスタンスタイプを含める必要はありません。ヘルプについては、[Amazon EC2 User Guide] (Amazon EC2 ユーザーガイド) 内の「[起動テンプレートの作成](#)」を参照してください。

Note

で起動テンプレートを使用する前にAmazon GameLift ServersFleetIQ、まず Auto Scaling グループを設定して、テンプレート設定と AMI が適切にデプロイされていることを確認することを強くお勧めします。

7. Amazon GameLift ServersFleetIQホスティングリソースをセットアップします。

ゲームサーバーをデプロイする各リージョンで、[CreateGameServerGroup\(\)](#) を呼び出してゲームサーバーグループを作成します。起動テンプレート (カスタム AMI とネットワークとセキュリティ設定を含む)、IAM ロール、およびゲームを実行できるインスタンスタイプのリストを渡します。このアクションは、変更Amazon GameLift ServersFleetIQできる Auto Scaling グループをアカウントにセットアップします AWS。その他のガイダンスと例については、「[Amazon GameLift ServersFleetIQ ゲームサーバーグループの管理](#)」を参照してください。

8. Amazon GameLift ServersFleetIQをゲームクライアントに統合します。

ゲームサーバー容量を割り当てるゲームクライアント、マッチメーカー、またはその他のバックエンドコンポーネントに AWS SDK を追加します。ゲームのタイプに応じて、マッチメーカーは [ListGameServers\(\)](#) または [ClaimGameServer\(\)](#) を呼び出して、サーバーの容量を取得し、利用可能なゲームサーバーを予約することができます。その他のガイダンスと例については、「[ゲームクライアントAmazon GameLift ServersFleetIQに統合する](#)」を参照してください。

9. Auto Scaling グループをスケールアップする。

Auto Scaling グループでインスタンスがプロビジョニングされると、インスタンスはゲームサーバーを起動します。その後、各ゲームサーバーは 使用可能な容量Amazon GameLift ServersFleetIQとして登録し、後でマッチメーカーによって一覧表示または要求されます。

10. ゲームをテストする。

マッチメーカーを呼び出し、`ClaimGameServer` を呼び出して、サーバーの容量を要求します。結果の IP とポートをゲームクライアントに渡して、ゲームサーバーに接続できるようにします。

Amazon GameLift ServersFleetIQ ゲームサーバーグループの管理

このトピックでは、Amazon GameLift ServersFleetIQゲームサーバーグループの設定に必要なタスクについて説明します。ゲームサーバーグループを作成すると、必要なすべての設定と、ゲームホス

ティング Amazon GameLift Servers FleetIQ の最適化を管理するための設定を含む EC2 Auto Scaling グループの作成がトリガーされます。

ゲーム サーバグループを作成する前に、少なくとも以下のリソースを準備しておく必要があります。

- ゲームサーバービルドで Amazon EC2 インスタンスを起動する方法を指定する Amazon EC2 起動テンプレート。詳細については、「Amazon EC2 ユーザーガイド」の「[ローンチテンプレートからのインスタンスの起動](#)」を参照してください。
- AWS アカウントへの制限付きアクセスを拡張して、Amazon GameLift Servers FleetIQ が Auto Scaling グループを作成して操作できるようにする IAM ロール。詳細については、「[サービス間でやり取りするための IAM ロールの作成](#)」を参照してください。

ゲーム サーバグループを作成する

ゲームサーバーグループを作成するには [CreateGameServerGroup\(\)](#) を呼び出します。このオペレーションでは、Amazon GameLift Servers FleetIQ ゲームサーバーグループと対応する Auto Scaling グループの両方が作成されます。ゲームサーバーグループを作成するときは、バランシング戦略やインスタンスタイプ定義など Amazon GameLift Servers FleetIQ、 のゲーム固有の設定を指定します。Auto Scaling グループの初期プロパティ設定も指定します。

次の例では、GameServerGroupの作成をトリガーします。これはc4.ラージ および c5.ラージ インスタンスタイプを指定し、グループをスポットインスタンスのみに制限します。Auto Scaling グループは、インスタンスをデプロイするために指定された起動テンプレートを使用し、ターゲット追跡 オートスケーリング ポリシーを使用して最小および最大設定内のグループキャパシティーを管理します。短いプロビジョニング期間が経過すると、AutoScalingGroup リソースが作成され、GameServerGroup が ACTIVE 状態になります。

```
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 \  
  --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}}'
```

ゲーム サーバークラスを更新する

リソースタイプの最適化など、ゲームサーバーのホスティング Amazon GameLift Servers FleetIQ を管理する方法に影響するゲームサーバークラスのプロパティを更新できます。プロパティを更新するには、[UpdateGameServerGroup](#) (更新ゲームサーバークラス) を呼び出します。ゲームサーバークラスへの変更が有効になると、Amazon GameLift Servers FleetIQ は Auto Scaling グループ内の特定のプロパティを上書きする可能性があります。

その他のすべての Auto Scaling グループプロパティ (MinSize、MaxSize、および LaunchTemplate など) は、Auto Scaling グループで直接変更できます。

以下の例では、インスタンスタイプ定義が c4.xlarge および c5.xlarge インスタンスタイプに切り替わるよう更新されています。

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

ゲーム サーバークラスインスタンスの追跡

デプロイされたインスタンスでゲームサーバークラスと Auto Scaling グループが起動して実行されると、[DescribeGameServerInstances\(\)](#) を呼び出して、ゲームサーバーインスタンスのステータスを追跡できます。このオペレーションを使用して、インスタンスのステータスを追跡できます。ゲームサーバークラスのステータスの詳細については、「[ゲーム サーバークラスの寿命](#)」を参照してください。

ゲームサーバークラスの下にある [Amazon GameLift Servers コンソール](#) を使用して、ゲームサーバークラスのステータスを監視することもできます。

ゲームサーバー Amazon GameLift Servers FleetIQ に統合する

このトピックでは、ゲームサーバープロジェクトと通信するための準備に必要なタスクについて説明します Amazon GameLift Servers FleetIQ。 「[Amazon GameLift Servers FleetIQ ベストプラクティス](#)」の追加ガイダンスを参照してください。

ゲームサーバーを登録する

ゲームサーバープロセスが起動され、ライブゲームプレイをホストする準備ができたら、[RegisterGameServer\(\)](#) を呼び出し Amazon GameLift ServersFleetIQ に登録する必要があります。登録すると Amazon GameLift ServersFleetIQ は、サーバー容量に関する情報をリクエストしたり、ゲームサーバーをリクエストしたりするときに、マッチメイキングシステムやその他のクライアントサービスに応答できます。登録時に、ゲームサーバーは、インバウンドクライアント接続に使用するポートと IP アドレスなど、Amazon GameLift ServersFleetIQ 関連するゲームサーバーデータと接続情報を提供できます。

```
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\"}"
```

ゲームサーバーのステータスを更新する

ゲームサーバーが登録されたら、サーバー容量の状態を同期させるために、定期的にヘルスと使用状況を報告する必要があります Amazon GameLift ServersFleetIQ。ヘルスと使用率のステータスを報告するには、[\[UpdateGameServer\]](#) (更新ゲームサーバー) を呼び出します。以下の例では、ゲームサーバーが正常であり、まだホスティングプレイヤーやゲームプレイによって占有されていないことが報告されています。

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

[Health status] (ヘルスステータス)

ゲームサーバーにヘルスステータスを追跡するメカニズムがある場合は、このメカニズムを使用してゲームサーバーのヘルス更新をトリガーできます Amazon GameLift ServersFleetIQ。

[Utilization status] (使用率ステータス)

ゲームサーバーの使用状況のレポートでは、どのゲームサーバーが現在最適で、新しいゲームセッションに使用できるかが Amazon GameLift ServersFleetIQ 常に把握されます。ゲームサーバーには、

使用率ステータスの更新をトリガーするメカニズムが必要ですAmazon GameLift ServersFleetIQ。たとえば、プレイヤーがゲームサーバーにConnectしたり、ゲームセッションの開始時に更新をトリガーしたりできます。

ゲームセッションを開始するとき、クライアントまたはマッチメイキングサービスは、利用可能なゲームサーバーをクレームし ([ClaimGameServer](#)) (クレームゲームサーバー) を呼び出して、プレイヤーにゲームサーバーへのConnectを促し、ゲームサーバーをトリガーしてゲームプレイをスタートします。このプロセスは、「[ゲームクライアントAmazon GameLift ServersFleetIQに統合する](#)」で説明されています。ゲームサーバーの「クレーム」は 60 秒間有効であり、ゲームサーバーはこのウィンドウ内で使用状況を更新できる必要があります。使用率ステータスが更新されない場合、はクレームAmazon GameLift ServersFleetIQを削除し、ゲームサーバーが使用可能であると仮定し、別のクライアントクレームリクエストのためにゲームサーバーを予約することがあります。

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

ゲームサーバーの登録を解除する

ゲームが終了すると、ゲームサーバーは [DeregisterGameServer\(\)](#) Amazon GameLift ServersFleetIQ の使用から登録を解除する必要があります。

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

ゲームクライアントAmazon GameLift ServersFleetIQに統合する

このトピックでは、ゲームセッションをホストするゲームサーバーを取得Amazon GameLift ServersFleetIQするために、と通信するためのゲームクライアントまたはマッチメイキングサービスを準備するために必要なタスクについて説明します。

ゲームクライアントまたはマッチメーカーがプレイヤーのゲームサーバーリソースをリクエストできるようにするメソッドを作成します。作成する方法にはいくつかのオプションがあります。

- 使用可能なゲームサーバーAmazon GameLift ServersFleetIQを選択します。このオプションは、低コストのスポットインスタンスと自動スケーリングを使用するためのAmazon GameLift ServersFleetIQ最適化を活用します。
- 利用可能なすべてのゲームサーバーをリクエストし、使用するサーバーを選択します (「リスト & ピック」とも呼ばれます)。

トピック

- [ゲームサーバーAmazon GameLift ServersFleetIQを選択](#)
- [自分のゲームサーバーを選択する](#)

ゲームサーバーAmazon GameLift ServersFleetIQを選択

で使用可能なゲームサーバーAmazon GameLift ServersFleetIQを選択するには、ゲームサーバーIDを指定せずに [ClaimGameServer\(\)](#) を呼び出します。このシナリオでは、Amazon GameLift ServersFleetIQは、ゲームホスティングに有効で、自動スケーリングに最適化されたインスタンス上のゲームサーバーを見つけるロジックを実行します。

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

クレームリクエストに応じて、は、クライアントがゲームサーバーへの接続に使用できるGameServerリソース、接続情報、ゲームデータAmazon GameLift ServersFleetIQを識別します。ゲームサーバーの要求ステータスは、60 秒間 [CLAIMED (クレーム済み)] に設定されます。プレイヤーの接続またはゲームプレイの開始Amazon GameLift ServersFleetIQ後に、ゲームサーバーまたはクライアントサービスのいずれかでのゲームサーバーのステータスを更新する必要があります。これにより、Amazon GameLift ServersFleetIQはゲームサーバー容量に対する後続のリクエストに応じてこのゲームサーバーを提供しません。 [\[UpdateGameServer\]](#) (更新ゲームサーバー) を呼び出してゲームサーバーのステータスを更新します。

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

自分のゲームサーバーを選択する

「リスト&ピック」メソッドを使用するには、[\[ListGameServers\]](#) (リストゲームサーバー) を呼び出して、ゲームクライアントまたはマッチメーカーが利用可能なゲームサーバーのリストを要求するようにします。ゲームサーバーのデータを使用して、プレイヤーやマッチメーカーがゲームサーバーを選択する際に使用できる追加情報を提供することができます。ページ分割された結果をリクエストし、ゲームサーバーを登録日順にソートし、結果を返す方法を制御できます。次のリクエストは、指定されたゲームサーバーグループ内のアクティブで利用可能な20台のゲームサーバーを、最新のゲームサーバーが最初に来るように登録時間順にソートされた状態で返します。

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

利用可能なゲームサーバーのリストに基づいて、クライアントまたはマッチメイキングサービスはゲームサーバーを選択し、特定のゲームサーバー ID付きの[\[ClaimGameServer\]](#) (クレームゲームサーバー) を呼び出してクレームします。このシナリオでは、Amazon GameLift ServersFleetIQは、「」で説明されているように、インスタンスタイプ最適化ロジックを実行しません[Amazon GameLift ServersFleetIQ ロジック](#)。

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

Amazon CloudWatch Amazon GameLift Servers FleetIQによるモニタリング

Amazon CloudWatch メトリクスを使用して、インスタンス容量のスケーリング、オペレーションダッシュボードの構築、アラームのトリガーを行います。はスタンドアロンソリューション Amazon GameLift Servers FleetIQとして、Amazon CloudWatch メトリクスのセットを AWS アカウントに発行します。また、[Amazon EC2 Auto Scaling ユーザーガイド](#)の Amazon CloudWatch を使用した Auto Scaling グループとインスタンスのモニタリングも参照してください。

FleetIQ メトリクスはここに一覧表示されます。メトリクス Amazon GameLift Serversでの完全な Amazon CloudWatch [Amazon GameLift Serversメトリクス](#)情報を参照してください。

メトリクス	説明
AvailableGameServers	<p>ゲーム実行に使用可能で、ゲームプレイに現在使用されていないゲームサーバーの数。この数には、クレーン済みであるが、まだ AVAILABLE ステータスにあるゲームサーバーが含まれます。</p> <p>単位: カウント</p> <p>関連する Amazon CloudWatch 統計: Sum</p> <p>ディメンション: GameServerGroup</p>
UtilizedGameServers	<p>ゲームプレイに現在使用されているゲームサーバー。この数字には、[UTILIZED (使用中)] 状態のゲームサーバーも含まれます。</p> <p>単位: カウント</p> <p>関連する Amazon CloudWatch 統計: Sum</p> <p>ディメンション: GameServerGroup</p>
DrainingAvailableGameServers	<p>ゲームプレイを現在サポートしていない、終了予定のインスタンスのゲームサーバー。これらのゲームサーバーは、新しいクレーンリクエストがあった場</p>

メトリクス	説明
	<p>合に、クレーンされる優先度が最も低いゲームサーバーです。</p> <p>単位: カウント</p> <p>関連する Amazon CloudWatch 統計: Sum</p> <p>ディメンション: GameServerGroup</p>
DrainingUtilizedGameServers	<p>ゲームプレイを現在サポートしている、終了予定のインスタンスのゲームサーバー。</p> <p>単位: カウント</p> <p>関連する Amazon CloudWatch 統計: Sum</p> <p>ディメンション: GameServerGroup</p>
PercentUtilizedGameServers	<p>ゲーム実行を現在サポートしているゲームサーバーの割合。このメトリクスは、ゲームサーバーキャパシティの現在の使用量を示します。プレイヤーの需要に合わせてインスタンスを動的に追加および削除できる Auto Scaling ポリシーを実行する場合に役立つメトリクスです。</p> <p>単位: パーセント</p> <p>関連する Amazon CloudWatch 統計: Average、Minimum、Maximum</p> <p>ディメンション: GameServerGroup</p>

メトリクス	説明
GameServerInterruptions	<p>スポットの可用性が制限されているために中断されたスポットインスタンスのゲームサーバー。</p> <p>単位: カウント</p> <p>関連する Amazon CloudWatch 統計: Sum</p> <p>ディメンション: GameServerGroup、InstanceType</p>
InstanceInterruptions	<p>可用性が制限されているために中断されたスポットインスタンス。</p> <p>単位: カウント</p> <p>関連する Amazon CloudWatch 統計: Sum</p> <p>ディメンション: GameServerGroup、InstanceType</p>

によるセキュリティ Amazon GameLift ServersFleetIQ

Amazon EC2 のスタンドアロン機能Amazon GameLift ServersFleetIQとして を使用している場合は、[Amazon EC2 ユーザーガイド](#)の「[Amazon EC2 のセキュリティ](#)」を参照してください。

Amazon EC2

のクラウドセキュリティが最優先事項 AWS です。AWS のユーザーは、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを利用できます。

セキュリティは、AWS とユーザーの間で共有される責任です。の使用時に責任共有モデルを適用する方法についてはAmazon GameLift ServersFleetIQ、「[のセキュリティ Amazon GameLift Servers](#)」を参照してください。

Amazon GameLift ServersFleetIQ リファレンスガイド

このセクションには、で使用するリファレンスドキュメントが含まれていますAmazon GameLift ServersFleetIQ。

トピック

- [Amazon GameLift ServersFleetIQ サービス API リファレンス \(AWS SDK\)](#)
- [Amazon GameLift ServersFleetIQ リリースノートと SDK バージョン](#)
- [Amazon GameLift Servers デベロッパーリソース](#)

Amazon GameLift ServersFleetIQ サービス API リファレンス (AWS SDK)

このトピックでは、の API アクションのタスクベースのリストを提供しますAmazon GameLift ServersFleetIQ。Amazon GameLift ServersFleetIQ サービス API は、`aws.gamelift`名前空間の AWS SDK にパッケージ化されます。[AWS SDK をダウンロードするか](#)、[Amazon GameLift ServersAPI リファレンスドキュメントを参照してください](#)。

Amazon GameLift ServersFleetIQ は、Amazon EC2 でのクラウドベースのゲームホスティングに低コストのスポットインスタンスの使用を最適化します。他のAmazon GameLift Serversホスティングオプションの詳細については、[Amazon GameLift Servers 「デベロッパーガイド」](#)を参照してください。

トピック

- [Amazon GameLift ServersFleetIQ API アクション](#)
- [利用可能なプログラミング言語](#)

Amazon GameLift ServersFleetIQ API アクション

以下のオペレーションでは、Amazon EC2 および Auto Scaling グループと組み合わせて、ゲームサーバーグループやゲームサーバーなどのAmazon GameLift ServersFleetIQリソースを管理できます。

ゲーム サーバークラスの管理

これらのオペレーションを使用して、FleetIQゲームサーバーのデプロイを最適化して管理します。ゲームサーバークラスは、Amazon EC2 インスタンスでゲームサーバープロセスを起動する方法を制御し、および Auto Scaling グループをセットアップし、FleetIQ最適化を適用する方法を定義します。

- [CreateGameServerGroup](#) – 新しいゲームサーバークラスおよび対応する Auto Scaling グループを作成し、ゲームサーバーをホストするインスタンスの起動を開始します。CLI コマンド:[ゲームサーバークラスの作成](#)
- [ListGameServerGroups](#) – Amazon GameLift Serversリージョン内のすべてのゲームサーバークラスのリストを取得します。CLI コマンド:[ゲームサーバークラスのリスト化](#)
- [DescribeGameServerGroup](#) – ゲームサーバークラスのメタデータを取得します。CLI コマンド:[ゲームサーバークラスの説明](#)
- [UpdateGameServerGroup](#) – ゲームサーバークラスのメタデータを変更します。CLI コマンド:[ゲームサーバークラスの更新](#)
- [DeleteGameServerGroup](#) – ゲームサーバークラスを完全に削除し、関連するホスティングリソースのFleetIQアクティビティを終了します。CLI コマンド:[ゲームサーバークラスの削除](#)
- [ResumeGameServerGroup](#) – ゲームサーバークラスの中断されたFleetIQアクティビティを復元します。CLI コマンド:[ゲームサーバークラスの再開](#)
- [SuspendGameServerGroup](#) – ゲームサーバークラスのFleetIQアクティビティを一時的に停止します。CLI コマンド:[ゲームサーバークラスの停止](#)

ゲームサーバーの管理

これらのオペレーションを使用して、FleetIQゲームサーバーのデプロイを最適化して管理します。ゲームサーバークラスは、Amazon EC2 インスタンスでゲームサーバープロセスを起動する方法を制御し、および Auto Scaling グループをセットアップし、FleetIQ最適化を適用する方法を定義します。

- [RegisterGameServer](#) – 新しいゲームサーバーから を呼び出して、ゲームサーバーがゲームプレイをホストする準備ができていAmazon GameLift ServersFleetIQることを通知します。CLI コマンド:[ゲームサーバークラスの登録](#)
- [ListGameServers](#) – ゲームクライアントサービスから呼び出して、ゲームサーバークラスで現在実行されているすべてのゲームサーバーのリストを取得します。CLI コマンド:[ゲームサーバーのリスト化](#)

- [ClaimGameServer](#) – ゲームクライアントサービスから呼び出して、新しいゲームセッションをホストするゲームサーバーを見つけて予約します。CLI コマンド:[ゲームサーバーの要求](#)
- [DescribeGameServer](#) – ゲームサーバーのメタデータを取得します。CLI コマンド:[ゲームサーバーの説明](#)
- [UpdateGameServer](#) – ゲームサーバーのメタデータ、健全性ステータス、または利用状況ステータスを変更します。CLI コマンド:[ゲームサーバーの更新](#)
- [DeregisterGameServer](#) – 終了するゲームサーバーから を呼び出して、ゲームサーバーグループからゲームサーバーを削除するAmazon GameLift ServersFleetIQように促します。CLI コマンド:[ゲームサーバーの解除](#)

利用可能なプログラミング言語

をサポートする AWS SDK Amazon GameLift Serversは、次の言語で使用できます。開発環境のサポートの詳細については、各言語のドキュメントを参照してください。

- C++ ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- Java ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- .NET ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- Go ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- Python ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- Ruby ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- PHP ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))
- JavaScript/Node.js ([SDK ドキュメント](#)) ([Amazon GameLift Servers](#))

Amazon GameLift ServersFleetIQ リリースノートと SDK バージョン

Amazon GameLift Servers リリースノートにはFleetIQ、サービスに関連する新機能、更新、修正に関する詳細が記載されています。このページには Amazon GameLift Servers SDK バージョン履歴も含まれています。

Amazon GameLift Servers デベロッパーリソース

すべてのAmazon GameLift Serversドキュメントと開発者リソースを表示するには、[Amazon GameLift Servers「ドキュメント」](#)のホームページを参照してください。

AWS 用語集

最新の AWS 用語については、AWS の用語集 リファレンスの[AWS 用語集](#)を参照してください。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。