



FleetIQ Developer Guide

Amazon GameLift



Version

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon GameLift: FleetIQ Developer Guide

Copyright © 2024 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

Amazon GameLift FleetIQ とは	1
FleetIQ の仕組み	2
Amazon GameLift FleetIQ ロジック	2
主なリソースとコンポーネント	5
ゲームアーキテクチャ	7
オンプレミスホスティングの補完	7
ゲーム サーバーグループの寿命	9
ゲームサーバーの寿命	11
スポット再調整プロセス	14
ベストプラクティス	16
Amazon GameLift FleetIQ の機能	19
Amazon GameLift FleetIQ の料金	20
セットアップ	21
対応ソフトウェア	21
AWS アカウントを設定する	22
を作成する AWS アカウント	22
Amazon GameLift FleetIQ のユーザーアクセス許可の管理	24
クロスサービスインタラクションのIAMロールを作成する	29
FleetIQ のゲームの準備	36
統合ステップ	36
ゲーム サーバーグループの管理	38
ゲーム サーバーグループを作成する	39
ゲーム サーバーグループを更新する	40
ゲーム サーバーグループインスタンスの追跡	40
ゲームサーバーの統合	40
ゲームサーバーを登録する	41
ゲームサーバーのステータスを更新する	41
ゲームサーバーの登録を解除する	42
ゲームクライアントの統合	42
Amazon GameLift FleetIQ にゲームサーバーを選択させる	43
自分のゲームサーバーを選択する	43
CloudWatch を使用してモニタリングする	45
FleetIQ を使用したセキュリティ	48
Amazon GameLift FleetIQ リファレンス	49

サービスAPIリファレンス (AWS SDK)	49
Amazon GameLift FleetIQ APIアクション	49
利用可能なプログラミング言語	51
リリースノートとSDKバージョン	51
すべての Amazon GameLift ガイド	51
AWS 用語集	52
.....	liii

Amazon GameLift FleetIQ とは

Amazon GameLift FleetIQ 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。Amazon GameLift FleetIQ を使用すると、Amazon EC2 および Amazon EC2 Auto Scaling のホスティングリソースと直接連携しながら、Amazon GameLift 最適化を活用して、プレイヤーに安価で回復力のあるゲームホスティングを提供できます。Amazon EC2 スポットインスタンスは、急激な割引で提供されますが、可用性が変動し、[中断の可能性があるため、ゲームホスティングには通常利用できません](#)。Amazon GameLift FleetIQ はこれらの制限を大幅に軽減し、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に使用できます。

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

Amazon GameLift FleetIQ ゲームホスティングソリューションは、以下のゲームデベロッパー向けに設計されています。

- フルマネージド型の Amazon GameLift サービスではなく、既存の AWS デプロイがあるか、Amazon EC2 を直接使用したいと考えています。Amazon GameLift FleetIQ 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。Amazon GameLift FleetIQ を使用すると、オンプレミス容量を使用し、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。

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

- AWS スキルビルダーで Amazon GameLift FleetIQ for [Game Servers を使用する コースを受講して、ゲームに Amazon GameLift FleetIQ 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 FleetIQ 統合ステップ」](#) の手順に従います。

Amazon GameLift FleetIQ の仕組み

Amazon GameLift FleetIQ ソリューションは、Amazon EC2および 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フルセットを補完するゲームホスティングレイヤーです。Amazon GameLift FleetIQ は、ゲームホスティングに固有の一連の機能を提供するだけでなく、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追加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 サービスと統合できます。

Amazon GameLift FleetIQ を使用する場合は、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および Auto Scaling のリソースと設定を使用して Amazon GameLift FleetIQ ゲームサーバーグループを作成します。このアクションは、Amazon GameLift FleetIQ に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促します。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 リソースEC2と Auto Scaling リソースへのフルアクセスが可能になります。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環として、Amazon GameLift FleetIQ は特定の Auto Scaling グループプロパティ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 FleetIQ 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。

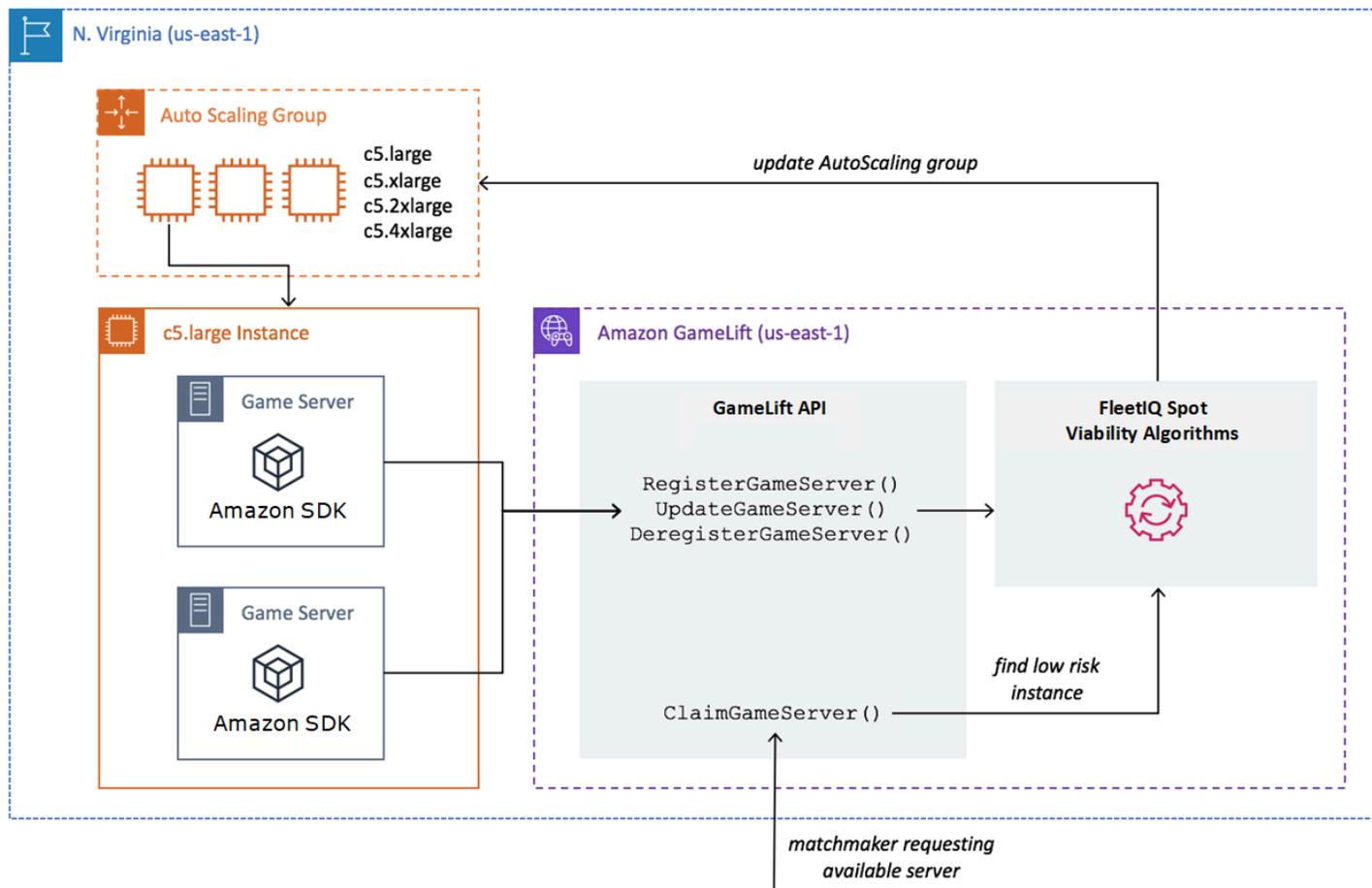
トピック

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

Amazon GameLift FleetIQ ロジック

次の図は、ゲームホスティングEC2のために Amazon と連携する際の Amazon GameLift FleetIQ の役割を示しています。その主な目的は、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

プレイ体験を提供するために最善のゲームサーバーを見つけることです。Amazon GameLift FleetIQ は、最適なリソースを、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リソースと定義します。Amazon GameLift FleetIQ は、この目標に 2 つの主要な方法でアプローチします。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。Amazon GameLift FleetIQ のジョブは、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確認し、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 FleetIQ は、Auto Scaling グループが次の方法で最適なインスタンスタイプを選択する方法に影響します。

- [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 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を与える バランシング戦略で設定されています。スポットインスタンスは、可用性の変動と潜在的な[中断](#)、Amazon GameLift FleetIQ がゲームサーバーホスティングのために最小限に抑える制限により、コストを削減します。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 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、Amazon GameLift FleetIQ は Auto Scaling グループの必要なインスタンスタイプのリストを継続的に更新し、現在実行可能なインスタンスタイプのみを含めます。
- これは、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。Amazon GameLift FleetIQ は、Auto Scaling グループ内の既存のインスタンスのうち、現在実行不可能なインスタンスタイプを識別します。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 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 FleetIQ 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。カスタムは、各インスタンスで新しいゲームサーバープロセスを起動および停止する方法AMIを制御します。新しいゲームサーバーが開始されると、Amazon GameLift FleetIQ に登録され、ゲームセッションをホストする準備ができています。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

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、Amazon GameLift FleetIQ に登録解除されます。

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またはその他のクライアントサービス) がゲームサーバーのリクエストを Amazon GameLift FleetIQ に送信します。Amazon GameLift FleetIQ は、利用可能なゲームサーバーを見つけ、新しいゲームセッションのためにクレームし、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、Amazon GameLift FleetIQ は、以下の意思決定プロセスを使用して、実行可能な低コストのスポットインスタンスで配置を最適化します。

1. 可能な場合、Amazon GameLift FleetIQ は、他のゲームセッションを既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 FleetIQ は、ドレイン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 FleetIQ は、実行可能なインスタンスで実行されている利用可能なすべての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

Amazon GameLift FleetIQ 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。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。の作成の詳細についてはAMI、[「Amazon 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。EC2AMIs はリージョン固有です。Amazon EC2ユーザーガイドの「[コピーAMIs](#)」で説明されているように、をあるリージョンAMIから別のリージョンにコピーできます。

- 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。を指定しAMI、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、その他のプロパティを設定します。起動テンプレートの作成の詳細については、「Amazon EC2ユーザーガイド」の「[起動テンプレートからインスタンスを起動する](#)」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。Amazon GameLift FleetIQ を使用する場合、Amazon GameLift FleetIQ が AWS アカウント内の Auto Scaling グループとEC2インスタンスリソースを作成およびアクセスできるようにする 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 FleetIQ 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っています。

- GameLift ゲームサーバーグループ。ゲームサーバーグループには、Amazon GameLift FleetIQ が対応する Auto Scaling グループ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、一部の設定を管理および変更するためのアクセス (IAMロールを引き受ける) があります。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、Amazon GameLift FleetIQ 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。
- GameLift ゲームサーバー。ゲームサーバーリソースは、Amazon GameLift FleetIQ 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 Amazon GameLift FleetIQ に登録し、それが属するゲームサーバーグループを識別するときに作成されます。Amazon GameLift FleetIQ 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を追跡するため、ゲームサーバーの可用性をモニタリングできます。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 FleetIQ リソースを介して作成されます。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 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、グループ容量を自動的にスケーリングします。Amazon GameLift FleetIQ では、ゲームサーバーグループと Auto Scaling グループの間に one-to-one 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、Amazon GameLift FleetIQ は、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るために、ロジックの一部として特定の設定を定期的に上書きおよび更新します。詳細については、「Amazon EC2 Auto Scaling ユーザーガイド [AutoScalingGroup](#)」の「」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 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。これらは通常、を持つ Auto Scaling グループによって起動されますAMI。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。Amazon GameLift FleetIQ を使用すると、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。

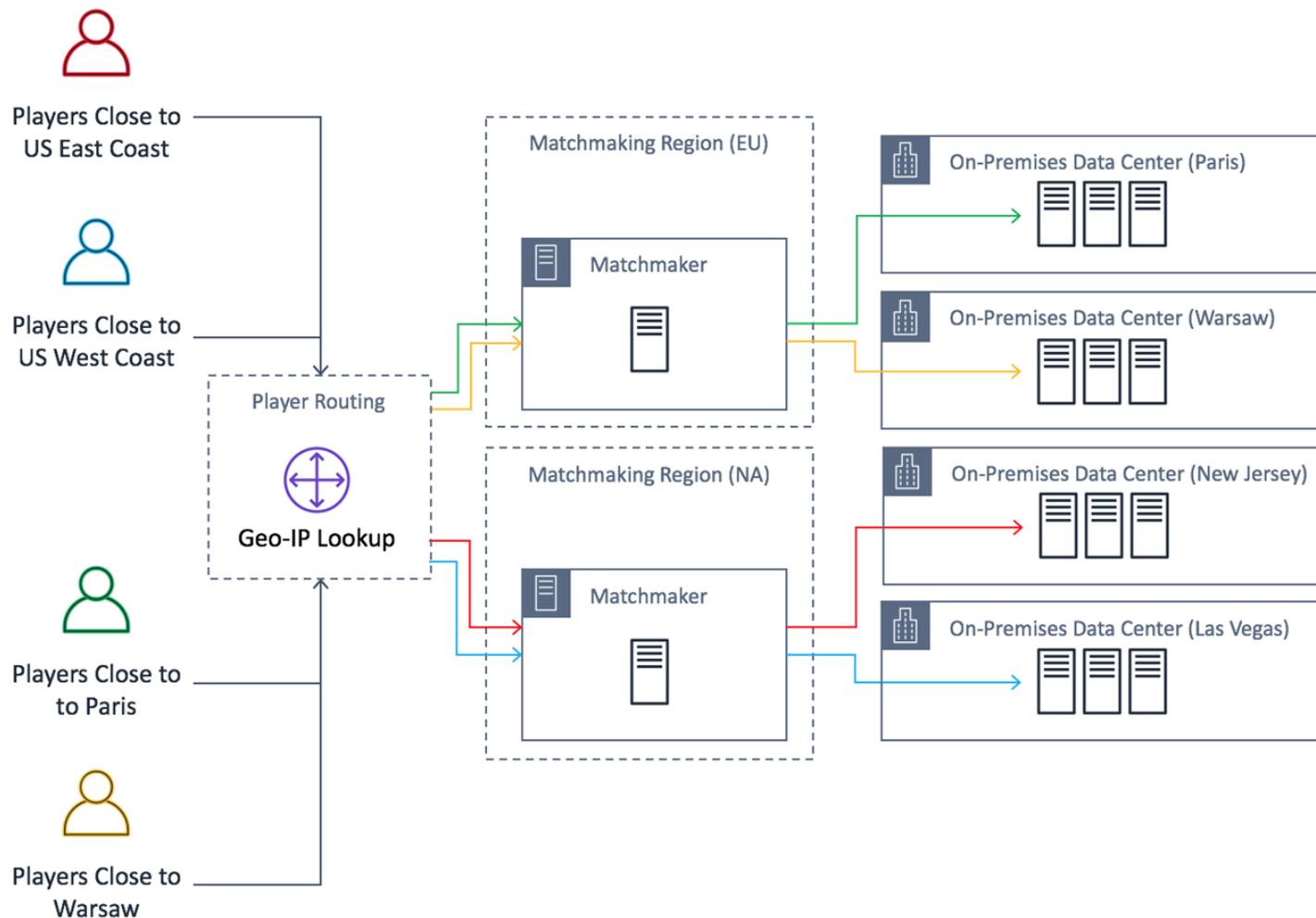
Amazon GameLift FleetIQ のゲームアーキテクチャ

オンプレミスホスティングの補完

Amazon GameLift FleetIQ は、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。これには、設定済みであるプレイヤーの geo-IP ルーティング、マッチメイキング、またはロビーサービスも含まれます。次の例は、Amazon GameLift FleetIQ 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。

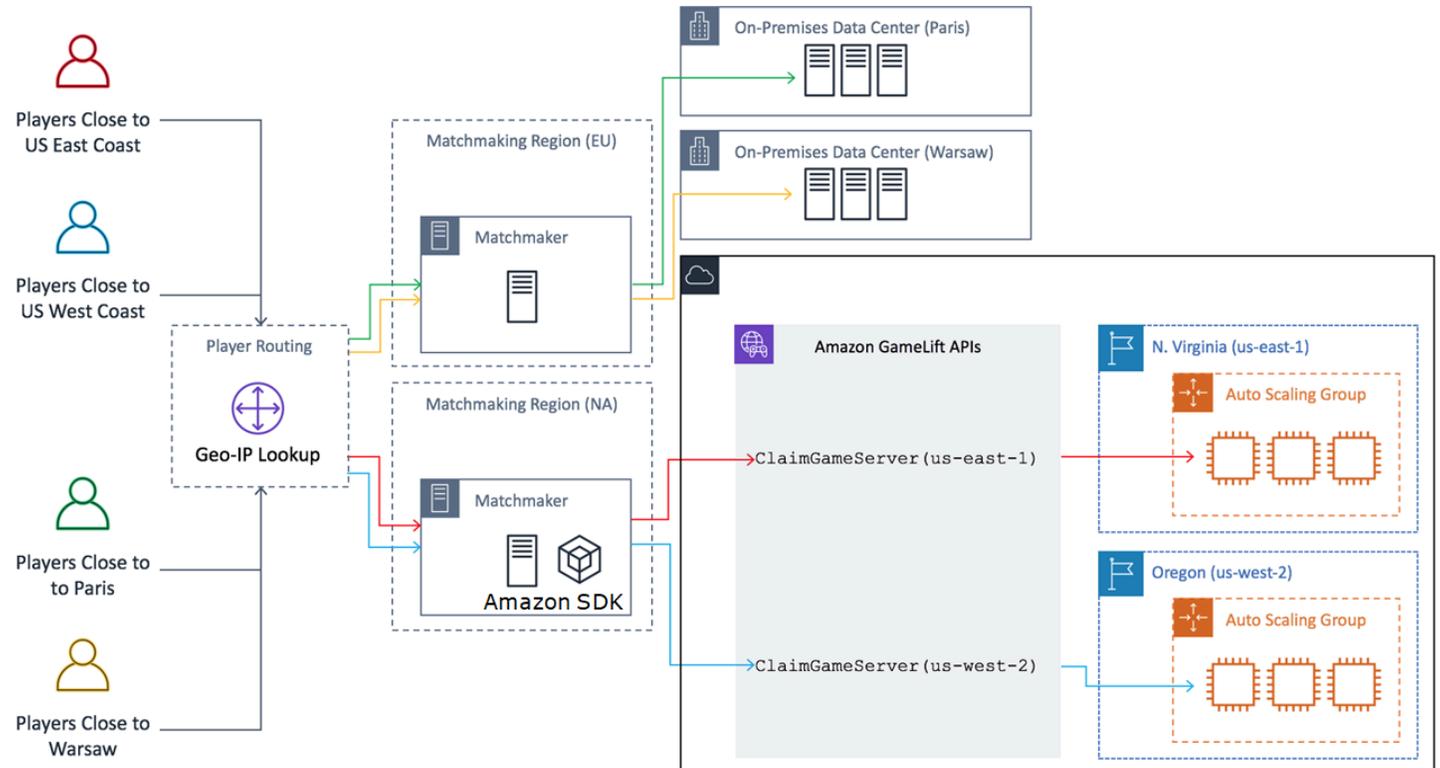


ゲームデベロッパーは、北米のゲームサーバーを Amazon GameLift FleetIQ が提供するサーバーに置き換える必要があります。スタートのために、ゲームサーバーを少し更新して Amazon GameLift FleetIQ の使用を有効にします。次に、Amazon マシンイメージ (AMI) を作成します。このイメージは、ゲーム用にデプロイされるすべての EC2 インスタンスにインストールされます。イメージには、ゲームサーバー、依存関係、およびプレイヤーのゲームセッションを実行するために必要なあらゆるものが含まれています。

AMI の準備ができたら、デベロッパーは Amazon GameLift FleetIQ ゲーム サーバークラスを各 AWS 北米 リージョン (us-east-1 と us-west-2) に 1 つずつ、2 つ作成します。デベロッパーは AMI を提供する起動テンプレート、必要なインスタンスタイプのリスト、グループのその他の設定を渡します。必要なインスタンスタイプのリストにより、ゲームホスティングに利用できるスポットインスタンスのチェックに使用するタイプを Amazon GameLift FleetIQ に指示します。

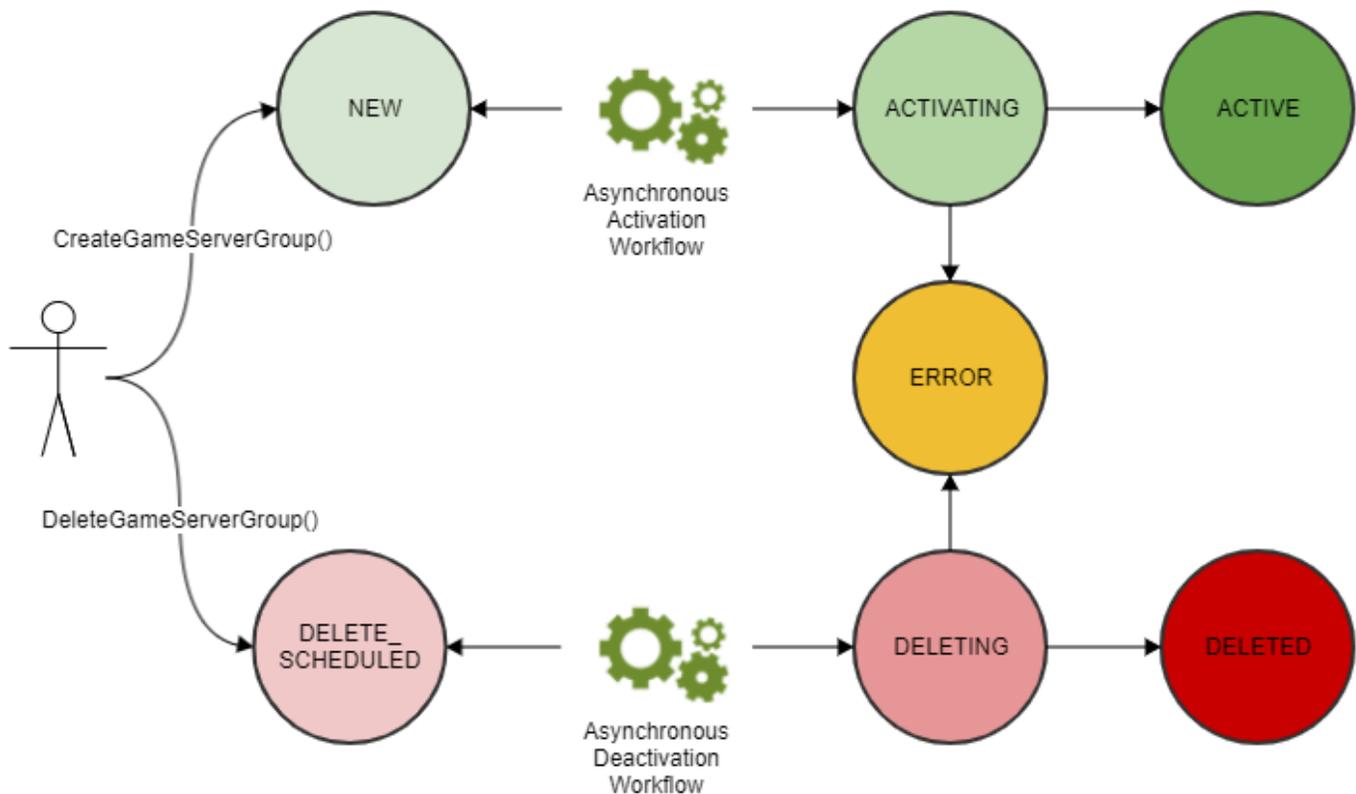
最後に、デベロッパーは AWS SDK と Amazon GameLift FleetIQ を北米のマッチメーカーに組み込み、新しいプレイヤーのグループがゲームセッションにサーバーキャパシティを必要とするときに

Amazon GameLift FleetIQ を呼び出します。Amazon GameLift FleetIQ は、使用可能なゲームサーバーがあるスポットインスタンスを検索し、プレイヤーのために予約して、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。新しいゲームをスタートする場合、プレイヤーはマッチメイキングを再入力します。これに応じて、Amazon GameLift FleetIQ は別の利用可能なゲームサーバーを見つけます。Amazon GameLift FleetIQ は、新しいゲームリクエストごとに反応し、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 FleetIQ は、スポットインスタンスの可用性が変動しても、ゲームホスティングに利用できないゲームサーバーから常にプレイヤーを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。



- Amazon GameLift API を呼び出し `CreateGameServerGroup()` で EC2 起動テンプレートと設定を渡すことで、ゲームサーバーグループを作成します。呼び出しに回答して、新しいゲームサーバーグループが作成され、ステータスに配置されます `NEW`。
- Amazon GameLift FleetIQ 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 `NEW` から `ACTIVATING` に移行します。ワークフローは、Amazon EC2 Auto Scaling グループや、提供されたAMIを持つEC2インスタンスを含む基盤となるリソースの作成を開始します。
- 何らかの理由でプロビジョニングが失敗した場合、ゲームサーバーグループはステータス `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はステータス `ACTIVE` に移行されます。この時点で、インスタンスは Amazon GameLift FleetIQ に登録されているゲームサーバーで起動されます。グループのインスタンスタイプは、ゲームホスティングに利用できるかどうか定期的に評価され、必要に応じて調整されます。Amazon GameLift FleetIQ 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も追跡し、ゲームサーバーのリクエストに対応します。

- 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。このアクションは、ゲームサーバーグループをステータス `DELETE_SCHEDULED` にします。削除をスケジュールできるのは、`ACTIVE` または `ERROR` 状態のゲームサーバーグループのみです。
- Amazon GameLift FleetIQ は、`DELETE_SCHEDULED` ステータスに回答し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。
- 何らかの理由で非アクティブ化が失敗した場合、ゲームサーバーグループはステータス `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。
- 非アクティブ化が成功すると、ゲームサーバーグループはステータス `DELETED` に移行されます。

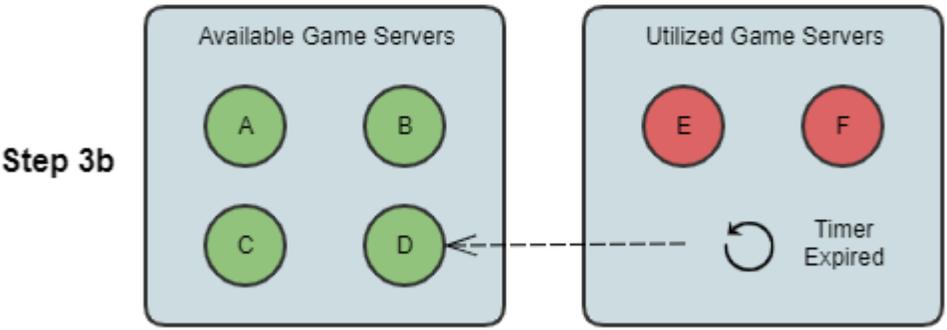
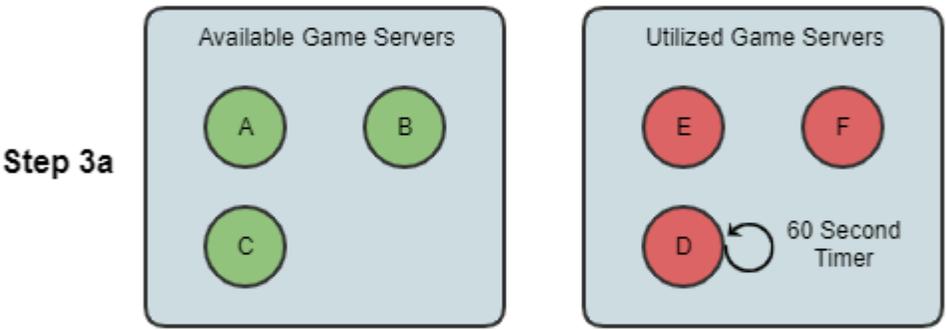
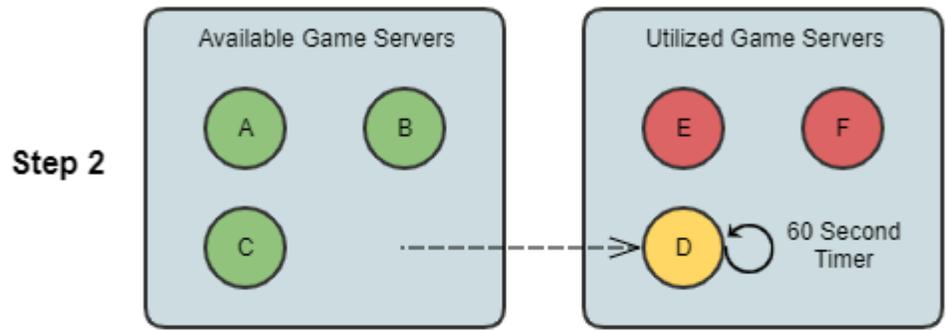
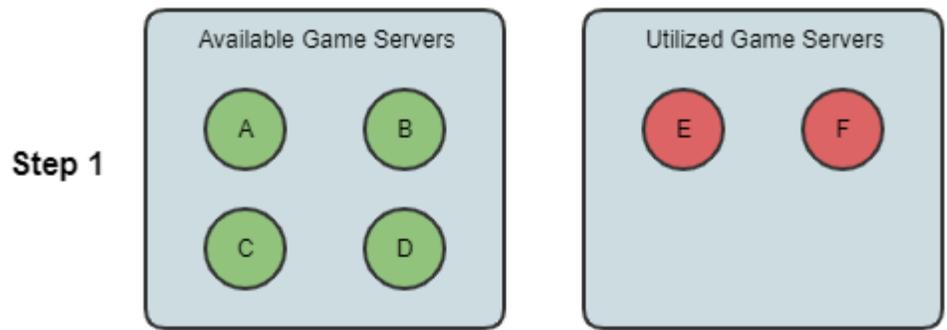
ゲームサーバーの寿命

Amazon GameLift FleetIQ を使用すると、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 FleetIQ にリンクされた 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 `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ることを Amazon GameLift FleetIQ に通知するときには作成されます。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ないことを示します。ゲームサーバーがゲームプレイで占有されると、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 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せん。

- 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。4 つは使用率ステータス AVAILABLE (A、B、C、D) で、2 つは現在 UTILIZED (E と F) です。
- Step 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 API を呼び出し `ClaimGameServer()` で新しいゲームサーバーをリクエストします。このリクエストにより、Amazon GameLift FleetIQ は利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 CLAIMED 秒間に設定するように求められます。Amazon GameLift FleetIQ 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応答します。ゲームサーバーでゲームプレイがまだ開始されていないため、その使用率ステータスは のままですが AVAILABLE、別のリクエストでリクエスト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、Amazon GameLift API を UTILIZED 呼び出して使用率ステータスを変更するには、60 秒以内にトリガーする必要があります `UpdateGameServer()`。
- 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 API を呼び出し `DeregisterGameServer()` で、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逸脱を Amazon GameLift FleetIQ に通知します。

スポット再調整プロセス

Amazon GameLift 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調整します。このプロセスは、ON_DEMAND_ONLY バランシング戦略を使用するゲームサーバーグループ、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。

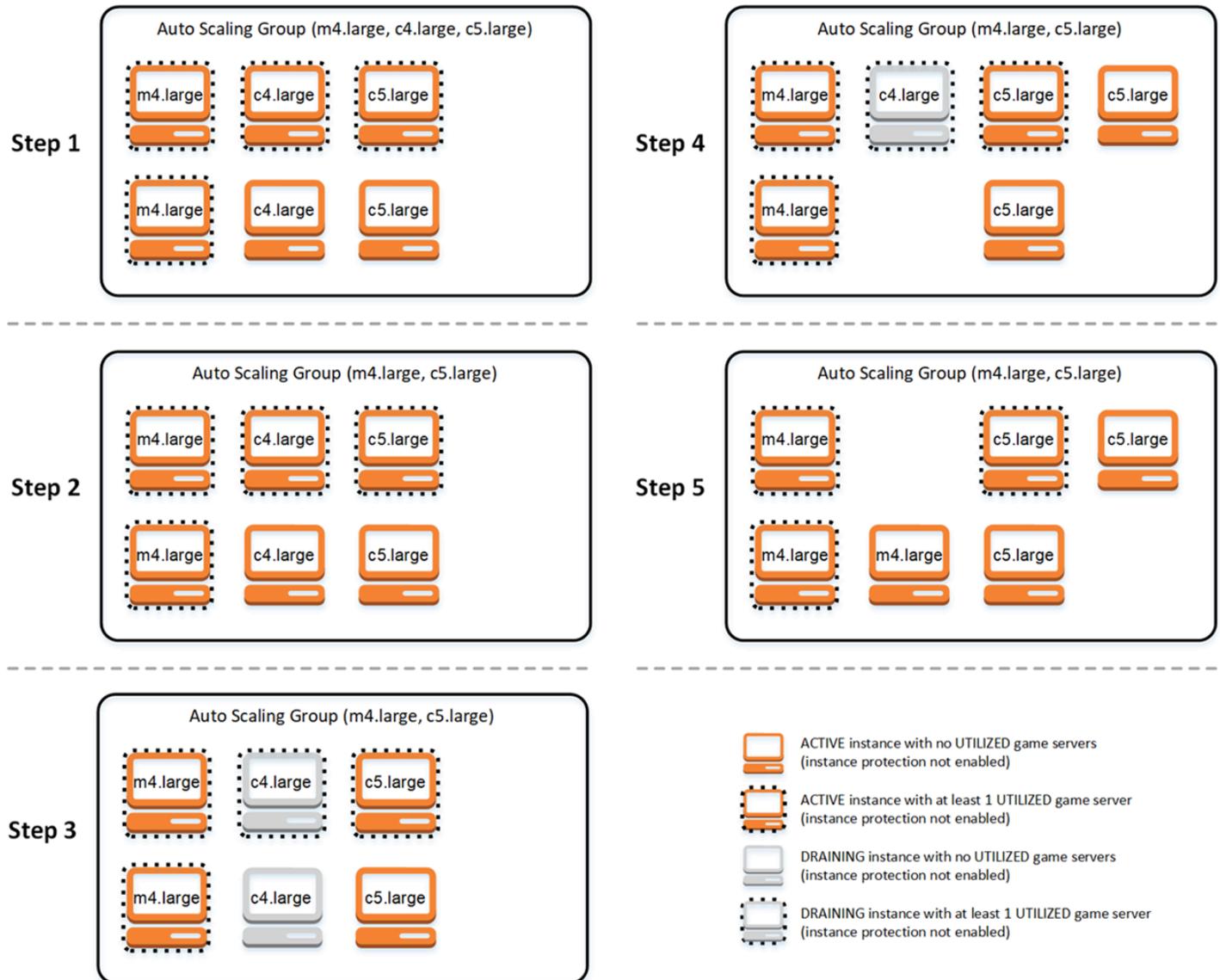
スポット再調整には、次の 2 つのキー目標があります。

- 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 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 つのインスタンスのステータスが少なくとも 1 つのゲームサーバー UTILIZED (破線の境界で表示) で、2 つのインスタンスは現在ゲームプレイをサポートしていません。

- Step 2. Amazon GameLift FleetIQ は、3 つのインスタンスタイプすべての現在のゲームホスティングの実行可能性を評価します。この評価では、c4.ラージ インスタンスタイプがゲームサーバーの中断の許容できない可能性を秘めていると判断されます。Amazon GameLift FleetIQ は Auto Scaling グループ設定をすぐに更新して、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 インスタンスの起動を防ぎます。
- ステップ 3 Amazon GameLift FleetIQ は c4.large タイプの既存のインスタンスを識別し、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 グループは必要な容量を維持し、グループが使用するインスタンスタイプの数は 3 つから 2 つに再調整されます。Amazon GameLift FleetIQ 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ます。

Amazon GameLift FleetIQ ベストプラクティス

Amazon GameLift FleetIQ は、ゲームホスティング用の Amazon EC2 リソースを管理するための低レベルのロジックレイヤーです。特に、Amazon GameLift FleetIQ 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、ゲームホスティングで実行可能なスポットインスタンスの使用を最適化します。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。

スタンドアロン機能としての Amazon GameLift FleetIQ では、フルマネージド Amazon GameLift ソリューションで提供されるアドバンス機能 (FleetIQ も使ってホスティングコストを最小化する) がありません。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のセッション管理、バージョニングなどの機能が必要な場合は、Amazon GameLift ソリューションをご覧ください。

Amazon GameLift FleetIQ を最大限に活用するために役立ついくつかのベストプラクティスを以下に紹介します。

- セッションベースのゲームに Amazon GameLift FleetIQ を使用する。Amazon GameLift FleetIQ は、ゲームセッションの中断の可能性が最も低いインスタンスにプレイヤーを常にルーティングする場合に最適です。寿命の長いセッションを保持すると、Amazon GameLift FleetIQ の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。Amazon GameLift FleetIQ は、セッションが 2 時間未満のゲームに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、Amazon GameLift FleetIQ で有効性の高いスポットインスタンスをゲームホスティングに利用するための柔軟性が増します。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、自動スケール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。
- Amazon GameLift FleetIQ を使用してゲームセッションを配置します。ゲームサーバーを使用するプレイヤーのグループを配置する場合は、Amazon GameLift API `ClaimGameServer()` を使用し

まず。Amazon GameLift FleetIQ は、ゲームセッションの中断の可能性が高いインスタンスへのプレイヤーの配置を回避します。

- Amazon GameLift FleetIQ にゲームサーバーのステータスを報告します。Amazon GameLift API `UpdateGameServer()` を使用して、サーバーのヘルスと使用率のステータスを定期的に報告します。ゲームサーバーの正確なステータスを維持することで、Amazon GameLift FleetIQ はゲームプレイをより効率的に配置できます。また、スポットのバ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 FleetIQ メトリクス `PercentUtilizedGameServers` は、現在使用されているホスティングキャパシティを示す指標です。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リングポリシー](#)」を参照してください。
- 開発環境と実稼働環境に異なる 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リングやバ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。
- EC2 でゲームをテストしてから、Amazon GameLift FleetIQ と統合してください。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。

以下の AWS ベストプラクティスも参照してください。

- [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 FleetIQ の機能

- [Optimized Spot balancing](最適化されたスポットバランシング) Amazon GameLift FleetIQ は、インスタンスタイプを定期的に評価し、ゲームセッションの中断の可能性があるために実行可能ではないと考えられる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に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 FleetIQ は、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 APIs 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。さらに、ゲームセッションは少ないインスタンスに集約されるため、不要なリソースをスケールダウンし、ホスティングコストを削減する EC2 Auto Scaling グループの能力が向上します。
- [Automatic scaling based on player usage](プレイヤーの使用状況に基づくオートスケーリング) Amazon GameLift 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。内の EC2 インスタンスと EC2 Auto Scaling グループを完全に制御します AWS アカウント。つまり、インスタンス起動テンプレートの設定、EC2 Auto Scaling グループ設定の維持、他の AWS サービスとの統合を行うことができます。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 FleetIQ はいくつかの 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 FleetIQ アクティビティを停止したりできます。
- [Support for multiple game server executable formats.](複数のゲームサーバーの実行可能形式の Support。) Amazon GameLift FleetIQ は、Windows EC2、Linux、コンテナ、Kubernetes など、現在 Amazon で実行されているすべての形式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リスト [EC2 については、「Amazon FAQs」](#) 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。Amazon GameLift FleetIQ を使用すると、ゲームサーバーホスティング用のさまざまな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)。つまり、ゲームサーバーを CPU、メモリ、ストレージ、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、プレイヤーに可能な限り最高のゲームエクスペリエンスを提供できます。
- Worldwide reach(ワールドワイドリーチ) Amazon GameLift FleetIQ 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 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 FleetIQ の料金

Amazon は、インスタンスの使用期間と転送されるデータ量別の帯域幅に対して GameLift 課金します。Amazon の料金と料金の完全なリストについては GameLift、「[Amazon の GameLift 料金](#)」を参照してください。

Amazon でゲームまたはマッチメイキングをホストするコストの計算については GameLift、「 の使用方法を説明する [GameLift」 Amazon 料金見積りの生成](#)」を参照してください [AWS Pricing Calculator](#)。

Amazon GameLift FleetIQ のセットアップ

このセクションのトピックでは、Amazon GameLift FleetIQ サービスで使用するための AWS アカウントの設定方法を含めた、タスクを設定する方法について説明します。

トピック

- [Amazon GameLift FleetIQ がサポートするソフトウェア](#)
- [Amazon GameLift FleetIQ の AWS アカウントを設定する](#)

Amazon GameLift FleetIQ がサポートするソフトウェア

Amazon GameLift FleetIQ は、Amazon でホストするための 64 ビット、マルチプレイヤーのゲームサーバー、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 EC2。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

Amazon GameLift FleetIQ は、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で使用できます EC2。これには、Amazon Linux、Ubuntu、Windows Server、Red Hat Enterprise Linux、SUSE Linux Enterprise Server、Fedora、Debian、CentOS、Gentoo Linux、Oracle Linux、および無料が含まれます BSD。現在の EC2 機能とサポートについては、「[Amazon EC2 の機能](#)」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する場合、Amazon GameLift FleetIQ は Kubernetes、Amazon Elastic Container Service (Amazon ECS)、および Amazon Elastic Kubernetes Service (EKS) との統合をサポートします EKS。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 FleetIQ サービスと通信するためある程度の統合が必要です。ゲームは を API 呼び出します AWS SDK。 [をダウンロードするか、AWS SDK Amazon GameLift API リファレンスドキュメントを参照してください。](#)

Amazon AWS SDK をサポートする GameLift は、以下の言語で利用できます。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](#))
- Java ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- .NET ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- Go ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- Python ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- Ruby ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- PHP ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- JavaScript/Node.js ([SDK ドキュメント](#)) ([Amazon GameLift](#))

Amazon GameLift FleetIQ の AWS アカウントを設定する

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

- Amazon GameLift FleetIQ で使用する AWS アカウントがまだない場合は、新しい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 FleetIQ 固有のアクセス許可を設定します。「[Amazon GameLift FleetIQ のユーザーアクセス許可の管理](#)」を参照してください。
- Amazon GameLift と Amazon EC2 リソースがやり取りできるようにする IAM ロールを作成します。「[クロスサービスインタラクションの IAM ロールを作成する](#)」を参照してください。

を作成する AWS アカウント

Amazon GameLift FleetIQ で使用する を作成してセットアップ AWS アカウント します。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/サインアップ> 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、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

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。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用して[ルートユーザーアクセスが必要なタスク](#)を実行してください。

AWS は、サインアッププロセスが完了した後に確認 E メールを送信します。/ に移動し、マイアカウント を選択すると、いつでも現在のアカウントアクティビティを表示<https://aws.amazon.com>し、アカウントを管理できます。

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、 のセキュリティを確保し AWS アカウントのルートユーザー、 を有効にして管理ユーザーを作成し AWS IAM Identity Center、 日常的なタスクにルートユーザーを使用しないようにします。

のセキュリティ保護 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、「[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 デバイスの有効化](#)」を参照してください。IAM

管理アクセスを持つユーザーを作成する

1. IAM Identity Center 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 Identity Center で、ユーザーに管理アクセスを許可します。

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的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 Identity Center ユーザーでサインインするには、IAM Identity Center ユーザーの作成時に E メールアドレスに URL 送信されたサインインを使用します。

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 「アクセスポータルへのサインイン」](#)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 Identity Center で、最小権限のアクセス許可を適用するベストプラクティスに従う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「[グループの参加](#)」を参照してください。

Amazon GameLift FleetIQ のユーザーアクセス許可の管理

必要に応じて、追加のユーザーを作成するか、Amazon GameLift FleetIQ アクセス許可を既存のユーザーに拡張します。Amazon GameLift FleetIQ 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は、これらのサービスにアクセスするアクセス許可を持っている必要があります。

ベストプラクティス ([のセキュリティのベストプラクティスIAM](#)) として、すべてのユーザーに最小権限のアクセス許可を適用します。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。

以下の手順に従って、AWS アカウントのユーザーを管理する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、ロールまたはユーザーグループに常にアクセス許可をアタッチします。

- [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。

- 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。

- ID プロバイダーIAMを介して で管理されるユーザー :

ID フェデレーションのロールを作成します。IAM ユーザーガイドの「[サードパーティー ID プロバイダーのロールの作成 \(フェデレーション\)](#)」の指示に従います。

- IAM ユーザー :

- ユーザーが担当できるロールを作成します。「[ユーザーガイド](#)」のIAM「[ユーザーのロールを作成する](#)」の手順に従います。IAM
- (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する。IAM ユーザーガイドの「[ユーザー \(コンソール\) へのアクセス許可を追加する](#)」の手順に従います。

リファレンス: Amazon GameLift FleetIQ _policy

以下は、参照用の Amazon GameLift FleetIQ _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 AWS CloudFormation アクセス許可を追加します。

```
{
    "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 外部とやり取りする場合は、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、 に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
<p>ワークフォースアイデンティティ</p> <p>(IAMアイデンティティセンターで管理されるユーザー)</p>	<p>一時的な認証情報を使用して AWS CLI、AWS SDKs、または へのプログラムによるリクエストに署名します AWS APIs。</p>	<p>使用するインターフェイス用の手引きに従ってください。</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「を使用する AWS CLI ようにを設定する AWS IAM Identity Center」を参照してください。 • AWS SDKs、ツール、および については AWS APIs、「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	<p>一時的な認証情報を使用して AWS CLI、AWS SDKs、または へのプログラムによるリクエストに署名します AWS APIs。</p>	<p>IAM「ユーザーガイド」のAWS「リソースで一時的な認証情報を使用する」の手順に従います。</p>
IAM	<p>(非推奨)</p> <p>長期認証情報を使用して、AWS CLI AWS SDKs、または へのプログラムによるリクエストに署名します AWS APIs。</p>	<p>使用するインターフェイス用の手引きに従ってください。</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。

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
		<ul style="list-style-type: none"> • および ツールについては AWS SDKs、AWS SDKs 「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。 • については AWS APIs、「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。 IAM

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。

クロスサービスインタラクションのIAMロールを作成する

Amazon GameLift FleetIQ 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、アカウントに IAM AWS 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ることで行われます。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。

- [Amazon GameLift FleetIQ のロールを作成する](#) Amazon EC2 リソースを更新するには。
- [Amazon のロールを作成する EC2](#) Amazon GameLift FleetIQ と通信するための リソース。

Amazon GameLift FleetIQ のロールを作成する

このロールにより、Amazon GameLift FleetIQ は スポット バランシング および 自動スケーリング 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 Amazon GameLift FleetIQ のロールを作成し、必要なアクセス許可を持つマネージド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 「サービスロールの作成」](#)と[AWS 「管理ポリシー」](#)を参照してください。

Console

これらのステップでは、 を使用して Amazon GameLift のマネージド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 AWS Management Console。

1. [IAM コンソール](#)を開き、`ロール: ロールの作成` を選択します。
2. 信頼できるエンティティの種類を選択 で、`AWS サービス` を選択します。
3. ユースケースを選択する では、`サービスのGameLiftリストから` を選択します。ユースケースの選択 で、適切な Amazon GameLift 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. アタッチされたアクセス許可ポリシーのリストには、1つのポリシー が含まれている必要があります `GameLiftGameServerGroupPolicy`。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 ロールのタグ付けやその他の用途の詳細については、「[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 「サービス: `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。
 - ポリシーには が含まれます `GameLiftGameServerGroupPolicy`。

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。ロールの概要ページの上部に表示されるロールARNの値に注意してください。Amazon GameLift FleetIQ ゲームサーバーグループを設定するときは、この情報が重要です。

AWS CLI

これらのステップでは、を使用して Amazon GameLift 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 AWS CLI。

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](#) で新しいIAMロールを作成し、作成した信頼ポリシーJSONファイルと関連付けます。

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。Amazon GameLift FleetIQ ゲームサーバーグループを設定するときは、この情報が必要です。

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

```
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、新しいロールの名前で [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。role-arn プロパティを新しいIAMロールARNの値に設定して [Gamelift create-game-server-group](#) を呼び出すことで、ロールが正しく設定されていることを確認できます。が ACTIVE状態GameServerGroupになると、Amazon GameLift FleetIQ がアカウント内の Amazon リソースEC2と Auto Scaling リソースを期待どおりに変更できることを示します。

Amazon のロールを作成する EC2

このロールにより、Amazon EC2リソースは Amazon GameLift FleetIQ と通信できるようになります。例えば、Amazon EC2インスタンスで実行されているゲームサーバーは、ヘルスステータスをレポートできる必要があります。Amazon GameLift FleetIQ 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。

AWS CLI を使用して Amazon のロールを作成し EC2、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、EC2を使用して Amazon のカスタム Amazon GameLift 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](#) を使用して新しいIAMロールを作成し、作成した信頼ポリシーJSONファイルと関連付けます。

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。Amazon EC2起動テンプレートを設定するときに、この情報が必要になります。

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、新しいロールの名前で [iam list-role-policies](#) を呼び出します。

5. Amazon で使用する新しいロールで [iam create-instance-profile](#) を使用してインスタンスプロファイルを作成しますEC2。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、ロールをインスタンスプロファイルにアタッチします。

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

これで、ロールとプロファイルを Amazon EC2起動テンプレートで使用できるようになりました。

Amazon GameLift FleetIQ 用のゲームの準備

このセクションでは、Amazon GameLift FleetIQ を使用して Amazon EC2 でゲームをホストするための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、次の操作を行う必要があります。

- ゲームサーバーを適応して Amazon GameLift FleetIQ と通信します。
- ゲームサーバーをデプロイする FleetIQ ゲームサーバーグループを作成します。
- 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 FleetIQ 統合ステップ](#)
- [Amazon GameLift FleetIQ ゲームサーバーグループの管理](#)
- [Amazon GameLift FleetIQ をゲームサーバーに統合する](#)
- [Amazon GameLift FleetIQ をゲームクライアントに統合する](#)

Amazon GameLift FleetIQ 統合ステップ

この統合プランは、Amazon GameLift FleetIQ 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稼働させるキーステップの概要を説明します。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 フラッグシップホスティングサービスを検索する場合、[Amazon GameLift 開発者ガイド](#)を参照してください。

Amazon GameLift FleetIQ の使用を開始するには、オンプレミスまたは Amazon EC2 環境のいずれかで動作す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 FleetIQ へのアクセス権を持つユーザーをセットアップします。

Amazon GameLift FleetIQ で使用するアカウントを新しい AWS アカウント を作成するか、既存のアカウントを選択します。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ための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、「[Amazon GameLift FleetIQ の AWS アカウントを設定する](#)」を参照してください。

2. IAM ロールを作成します。

Amazon GameLift FleetIQ、Amazon EC2 および Auto Scaling のリソースが相互に通信を許可できるようにロールを作成します。詳細については、「[クロスサービスインタラクションのIAMロールを作成する](#)」を参照してください。

3. Amazon GameLift FleetIQ 機能のある AWS SDK および AWS CLI を入手する。

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

4. Amazon GameLift FleetIQ で使用するようゲームサーバーを準備する。

AWS SDK をゲームサーバープロジェクトに追加し、Amazon GameLift FleetIQ をゲームサーバーの現在のステータスと使用状況を更新しておくためのコードを追加します。その他のガイダンスと例については、「[the section called “ゲームサーバーの統合”](#)」を参照してください。この情報を使用して、Amazon GameLift FleetIQ は、有効で未使用のゲームサーバーのリストをマッチメイキングシステムに提供し、また、プレイヤーが存在するインスタンスをバランシング中に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、ゲームサーバーが Amazon GameLift FleetIQ と通信するためのアクセス許可を使用して、作成したインスタンスプロファイル (ステップ 2 を参照) をリファレンスする必要があります。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

Amazon GameLift FleetIQ で起動テンプレートを使用する前に、まず Auto Scaling グループを設定して、テンプレート設定と AMI が正しくデプロイされていることを確認するようにしてください。

7. Amazon GameLift FleetIQ ホスティングリソースをセットアップする

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。このアクションでは、Amazon GameLift FleetIQ が修正できる AWS アカウントに Auto Scaling グループをセットアップします。その他のガイダンスと例については、「[Amazon GameLift FleetIQ 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 FleetIQ をゲームクライアントに統合する。

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、「[Amazon GameLift FleetIQ をゲームクライアントに統合する](#)」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、利用可能な容量として Amazon GameLift FleetIQ に登録されるか、後にマッチメーカーがリストまたはクレームします。

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 FleetIQ ゲームサーバーグループの管理

このトピックでは、Amazon GameLift FleetIQ ゲームサーバーグループのセットアップ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、ゲームホスティングの Amazon

GameLift FleetIQ 最適化を管理するための設定とともに、必要なすべての設定で 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「[起動テンプレートからインスタンスを起動する](#)」を参照してください。
- FleetIQ が Auto Scaling グループを作成して操作できるようにする GameLift、AWS アカウントへの制限付きアクセスを拡張する IAM ロール。詳細については、「[クロスサービスインタラクションの IAM ロールを作成する](#)」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 FleetIQ ゲームサーバ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 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を管理します。プロビジョニング期間が短いと、AutoScalingGroup リソースが作成され、は ACTIVE 状態 GameServerGroup 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 サーバーグループを更新する

リソースタイプの最適化など、Amazon GameLift FleetIQ がゲームサーバーのホスティングを管理する方法に影響するゲームサーバーグループのプロパティを更新できます。これらのプロパティを更新するには、[UpdateGameServerGroup\(\)](#) を呼び出します。ゲームサーバーグループの変更が有効になると、Amazon GameLift FleetIQ は Auto Scaling グループの特定のプロパティを上書きすることがあります。

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 サーバーグループインスタンスの追跡

インスタンスを作成してゲームサーバーグループと Auto Scaling グループにデプロイしたら、[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。ゲームサーバーグループのステータスの詳細については、「[ゲームサーバーグループの寿命](#)」を参照してください。

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

Amazon GameLift FleetIQ をゲームサーバーに統合する

このトピックでは、Amazon GameLift FleetIQ と通信するためのゲームサーバープロジェクトを準備するために必要なタスクについて説明します。「[Amazon GameLift 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なら、[RegisterGameServer\(\)](#) を呼び出して Amazon GameLift FleetIQ に登録する必要があります。登録すると、Amazon GameLift FleetIQ は、サーバー容量に関する情報のリクエスト時、またはゲームサーバーをリクエストする際に、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せん。登録時に、ゲームサーバーは Amazon GameLift FleetIQ に、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、関連するゲームサーバーの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、Amazon GameLift FleetIQ で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率のステータスを報告する必要があります。[UpdateGameServer\(\)](#) を呼び出して、ヘルスと使用率のステータスを報告します。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 Amazon GameLift FleetIQ へのゲームサーバーヘルス更新をトリガーできます。

[Utilization status] (使用率ステータス)

ゲームサーバー使用率ステータスを報告すると、Amazon GameLift FleetIQ 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を把握できます。ゲームサーバーには、Amazon GameLift FleetIQ の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。

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、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 ([ClaimGameServer\(\)](#) を呼び出して) 要求し、プレイヤーにゲームサーバーに接続するよう促し、ゲームサーバーをトリガーしてゲームプレイを開始します。このプロセスは、「[Amazon GameLift FleetIQ をゲームクライアントに統合する](#)」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、Amazon GameLift FleetIQ はクレームを削除し、ゲームサーバーが使用可能であることを前提とし、ゲームサーバーを別のクライアントクレームリクエスト用に予約できます。

```
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\(\)](#) を使用して Amazon GameLift FleetIQ 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
```

Amazon GameLift FleetIQ をゲームクライアントに統合する

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するために、Amazon GameLift FleetIQ 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。

- Amazon GameLift FleetIQ で利用可能なゲームサーバーを選択します。このオプションは、Amazon GameLift FleetIQ 最適化を利用して、低コストのスポットインスタンスと自動スケールリングを使用します。

- 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

- [Amazon GameLift FleetIQ にゲームサーバーを選択させる](#)
- [自分のゲームサーバーを選択する](#)

Amazon GameLift FleetIQ にゲームサーバーを選択させる

Amazon GameLift FleetIQ で使用可能なゲームサーバーを選択するには、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 FleetIQ は口ジックを実行して、ゲームホスティングに実行可能で、自動スケーリング用に最適化されたインスタンス上のゲームサーバーを検索します。

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

クレームリクエストに応じて、Amazon GameLift FleetIQ 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、ゲームデータを識別します。ゲームサーバーのクレームステータスは 60 CLAIMED 秒間 に設定されます。ゲームサーバーまたはクライアントサービスは、プレイヤーが接続またはゲームプレイを開始した後に、Amazon GameLift FleetIQ で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 FleetIQ は、ゲームサーバー容量に対する後続のリクエストに回答してこのゲームサーバーを提供しなくなります。[UpdateGameServer\(\)](#) 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

「list and pick」メソッドを使用すると、ゲームクライアントまたはマッチメーカーは [ListGameServers\(\)](#) を呼び出して利用可能なゲームサーバーのリストをリクエストします。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 で [ClaimGameServer\(\)](#) を呼び出して請求します。このシナリオでは、Amazon GameLift FleetIQ 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いずれも実行しません[Amazon GameLift FleetIQ ロジック](#)。

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

Amazon CloudWatch で Amazon GameLift FleetIQ をモニタリングする

Amazon CloudWatch メトリクスを使うことは、インスタンス容量のスケール、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に役立ちます。スタンドアロンソリューションとしての Amazon GameLift FleetIQ は、Amazon CloudWatch メトリクスのセットを AWS アカウントに出力します。また、Amazon EC2 Auto Scaling ユーザーガイドにある [Amazon CloudWatchを使った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。 [Amazon CloudWatch メトリクス](#) の Amazon GameLift の完全な Amazon CloudWatch 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>単位はカウント</p> <p>関連する 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>単位はカウント</p> <p>関連する 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、クレームされる優先度が最も低いゲームサーバーです。</p> <p>単位はカウント</p> <p>関連する 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>単位はカウント</p> <p>関連する 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 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>単位はカウント</p> <p>関連する 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>単位はカウント</p> <p>関連する CloudWatch 統計: Sum</p> <p>ディメンション: GameServerGroup、InstanceType</p>

Amazon GameLift FleetIQ でのセキュリティ

Amazon EC2 のスタンドアロン機能として Amazon GameLift FleetIQ を使用している場合は、[Amazon EC2 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。Amazon GameLift FleetIQ 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 [のセキュリティ GameLift](#)」を参照してください。

Amazon GameLift FleetIQ リファレンスガイド

このセクションでは、Amazon GameLift FleetIQ で使用するリファレンスドキュメントについて説明します。

トピック

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

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

このトピックでは、Amazon GameLift FleetIQ 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 FleetIQ サービスはAPI、`aws.gamelift`名前空間の AWS SDKにパッケージ化されます。[をダウンロードする AWS SDK](#)が、[Amazon GameLift API リファレンスドキュメント](#)を参照してください。

Amazon GameLift FleetIQ は、Amazon 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EC2。他の [Amazon ホスティングオプションの詳細については、「Amazon GameLift デベロッパーガイド」](#)を参照してください。 GameLift

トピック

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

Amazon GameLift FleetIQ APIアクション

以下のオペレーションでは、ゲームサーバーグループやゲームサーバーを含む Amazon GameLift FleetIQ リソースを Amazon EC2 および Auto Scaling グループと併せて管理できます。

ゲームサーバーグループの管理

これらの操作を使用して、FleetIQ 最適化でゲームサーバーのデプロイメントを管理します。ゲームサーバーグループは、Amazon EC2インスタンスでのゲームサーバープロセスの起動方法を制御し、および Auto Scaling グループを設定し、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グループと対応する Auto Scaling グループを作成し、ゲームサーバーをホストするためのインスタンスの起動を開始します。CLI コマンド: [create-game-server-group](#)
- [ListGameServerGroups](#) – Amazon GameLift リージョン内のすべてのゲームサーバーグループのリストを取得します。CLI コマンド: [list-game-server-groups](#)
- [DescribeGameServerGroup](#) – ゲームサーバーグループのメタデータを取得します。CLI コマンド: [describe-game-server-group](#)
- [UpdateGameServerGroup](#) – ゲームサーバーグループのメタデータを変更します。CLI コマンド: [update-game-server-group](#)
- [DeleteGameServerGroup](#) – ゲームサーバーグループを完全に削除し、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: [delete-game-server-group](#)
- [ResumeGameServerGroup](#) – ゲームサーバーグループの中断された FleetIQ アクティビティを復元します。CLI コマンド: [resume-game-server-group](#)
- [SuspendGameServerGroup](#) – ゲームサーバーグループの FleetIQ アクティビティを一時的に停止します。CLI コマンド: [suspend-game-server-group](#)

ゲームサーバーの管理

これらの操作を使用して、FleetIQ 最適化におけるゲームサーバーのデプロイメントを管理します。ゲームサーバーグループは、Amazon EC2 インスタンスでのゲームサーバープロセスの起動方法を制御し、および Auto Scaling グループを設定し、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ることを Amazon GameLift FleetIQ に通知します。CLI コマンド: [register-game-server-group](#)
- [ListGameServers](#) – ゲームクライアントサービスから を呼び出して、ゲームサーバーグループ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: [list-game-servers](#)
- [ClaimGameServer](#) – ゲームクライアントサービスから を呼び出して、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: [claim-game-server](#)
- [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: [describe-game-server](#)
- [UpdateGameServer](#) – ゲームサーバーのメタデータ、ヘルスステータス、または使用率ステータスを変更します。CLI コマンド: [update-game-server](#)

- [DeregisterGameServer](#) – 終了するゲームサーバーから呼び出して、Amazon GameLift FleetIQ に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ように促します。CLI コマンド: [deregister-game-server](#)

利用可能なプログラミング言語

Amazon をサポートする AWS SDK GameLift は、次の言語で利用できます。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](#))
- Java ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- .NET ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- Go ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- Python ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- Ruby ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- PHP ([SDK ドキュメント](#)) ([Amazon GameLift](#))
- JavaScript/Node.js ([SDK ドキュメント](#)) ([Amazon GameLift](#))

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

Amazon GameLift リリースノートには、サービスに関連する新しい FleetIQ の機能、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 SDK バージョン履歴も含まれています。

Amazon デ GameLift ベロツパーリソース

すべての Amazon GameLift ドキュメントと開発者リソースを表示するには、[Amazon GameLift 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

AWS の最新の用語については、「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。