



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 の商標とトレードドレスは、Amazon 以外の製品またはサービスとの関連において、顧客に混乱を招いたり、Amazon の名誉または信用を毀損するような方法で使用することはできません。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
主要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
ライフサイクルと状態	10
スポット再調整プロセス	10
ゲーム サーバーグループの寿命	12
ゲームサーバーの寿命	14
ベストプラクティス	17
Amazon GameLift FleetIQ の特徴	20
Amazon GameLift FleetIQ の料金	21
セットアップ	22
対応ソフトウェア	22
AWSアカウントの設定	23
AWS アカウント の作成	23
Amazon GameLift FleetIQ のユーザーアクセス許可を管理する	25
サービス間でやり取りするための IAM ロールの作成	30
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
Amazon GameLift FleetIQ リファレンス	48

サービス API リファレンス (AWS SDK)	48
Amazon GameLift FleetIQ API アクション	48
利用可能なプログラミング言語	50
FleetIQ を使用したセキュリティ	51
リリースノートと SDK バージョン	52
すべての Amazon GameLift ガイド	53
AWS 用語集	54
.....	iv

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向けに低コストの 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 GameLift の最適化を活用しながら Amazon EC2 と Amazon EC2 Auto Scaling のホスティングリソースを直接操作し、低コストで耐障害性の高い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、大幅な割引で提供されていますが、可用性が変動し、[\[interruptions\]](#) (中断) の可能性があるため、ゲームホスティングでは一般的に実行できません。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で使用できるようにします。

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場合にも使用できます。Amazon GameLift の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。

- 既存の AWS デプロイがある、またはフルマネージド Amazon GameLift サービスを経由するのではなく、Amazon EC2 を直接使用したい場合。Amazon GameLift FleetIQ は、AWS アカウントで管理している Auto Scaling グループで動作します。Amazon 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)、および AWS Shield Advanced といった他の AWS サービスとの統合も可能です。
- 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 を使うと、既存のオンプレミスキャパシティを活用するハイブリッドデプロイシステムを構築し、AWS クラウドのキャパシティを必要に応じて徐々に追加できます。

Amazon GameLift FleetIQ で作業をスタートする準備はできましたか

- AWS スキルビルダーの「[Amazon GameLift FleetIQ をゲームサーバーに使用する](#)」コースを受講して、ゲームに Amazon GameLift FleetIQ を使用する方法を学びましょう。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 で使用できるコンピューティングリソース管理ツール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特有の一連の機能のスレート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するためのロジックのレイヤーを追加で提供します。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。

Amazon GameLift FleetIQ を使用する場合は、Amazon EC2 インスタンスを通常通り起動する準備を行います。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のではなく、Amazon EC2 および Auto Scaling 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。このアクションにより、Amazon GameLift FleetIQ は、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。

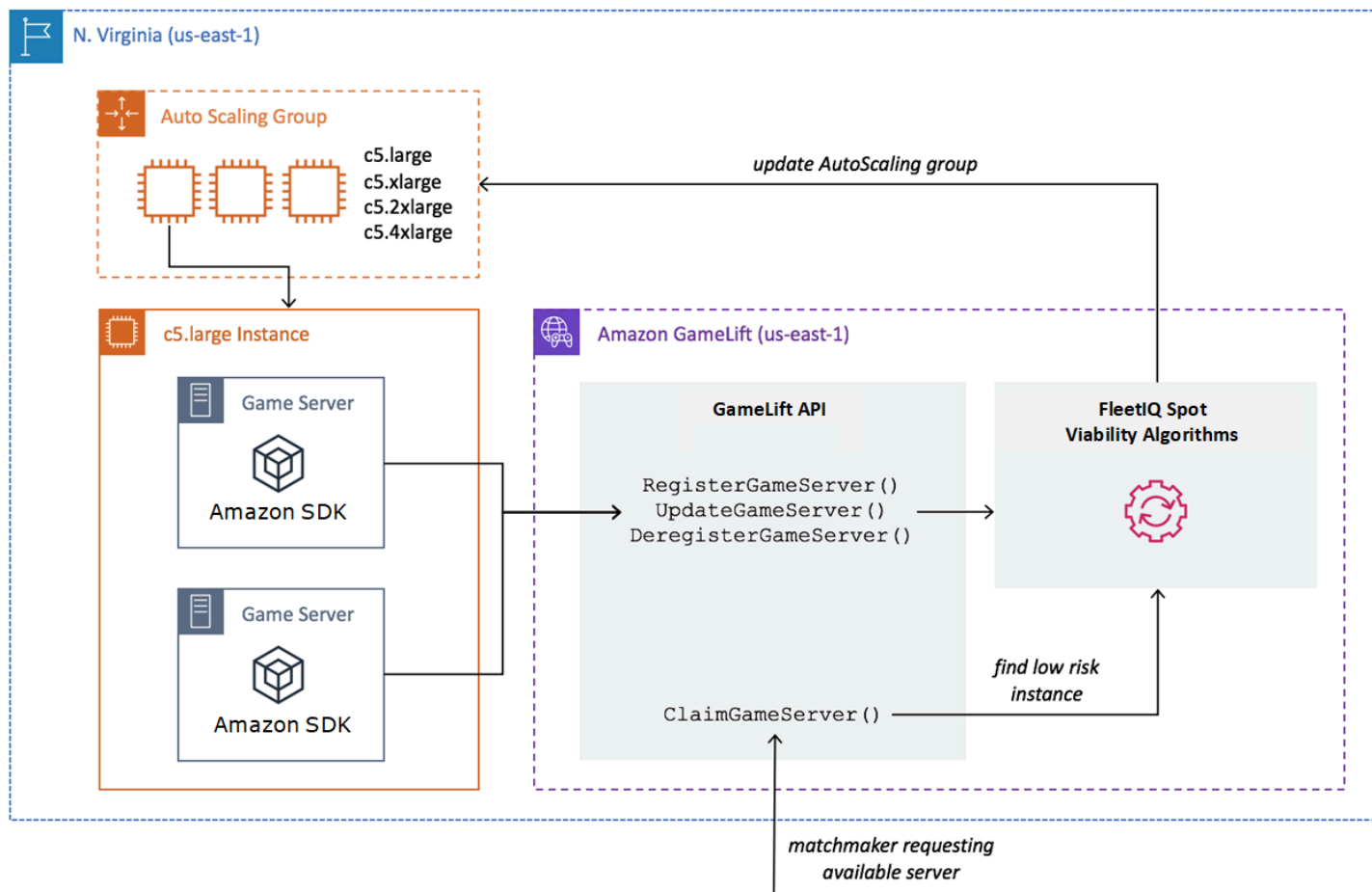
Auto Scaling グループが作成されると、Amazon EC2 リソースと Auto Scaling リソースへのフルアクセス権が付与されます。Auto Scaling グループの設定を変更したり、マルチレベル スケーリングポリシーやロードバランサーなどを追加したり、他の AWS のサービスと統合したりできます。グループ内のインスタンスに直接 Connect できます。Amazon GameLift FleetIQ は、最適化ロジックのパートとして、特定の Auto Scaling グループプロパティを定期的に更新します。Auto Scaling グループによってデプロイされたすべての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。

Amazon GameLift FleetIQ ロジック

次の図表は、ゲームホスティングを Amazon EC2 で操作する際の Amazon GameLift FleetIQ のロールを示しています。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、最低の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最高のリソースを定義します。Amazon GameLift FleetIQ は、2つのキーな方法でこの目標に取り組みます。1つは Auto Scaling グループで有効なインスタンスタイプのみを許可するという事。もう1つは、グループの利用可能なリソース全体で効果的に新しいゲームセッションを配置することです。



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 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 グループによるスポットインスタンスやオンデマンドインスタンスの使用 방법에影響するバランシング戦略が設定されています。スポットインスタンスは可用性の変動や [\[中断\]](#) の可能性があるため、低コストとなります。Amazon GameLift FleetIQ は、ゲームサーバーホスティングに対するこのような制限を最低限に抑えます。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のレート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各目的の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 グループの目的のインスタンスタイプのリストを継続的に更新して、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、現在無効なインスタンスタイプである Auto Scaling グループの既存インスタンスを特定します。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の最適なプレイメントを決定します。

Amazon GameLift FleetIQ によるゲームサーバーの追跡を有効にするには、ゲームサーバーソフトウェアがそのステータスを報告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を起動すると、それが 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整ったことを

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、Amazon GameLift FleetIQ が登録解除されます。

新しいゲームセッションをスタートする場合は、ゲームクライアント (またはマッチメーカーや他のクライアントサービス) からゲームサーバーのリクエストが Amazon GameLift FleetIQ に送信されます。Amazon GameLift FleetIQ は、利用可能なゲームサーバーを検索し、それを新しいゲームセッション用としてクレームし、ゲームサーバー ID と接続情報を返します。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 は次の決定プロセスを使用して、有効かつ低コストのスポットインスタンスのプレイスメントを最適化します。

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、ゲーム ホスティングに無効な [ドレイン中] フラグが立った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、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

Amazon GameLift FleetIQ でゲームホスティングリソースをセットアップする前に、AWS アカウントで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。AMI の作成の詳細については、「Linux インスタンス用 Amazon EC2 ユーザーガイド」内の[\[Amazon Machine Images\]](#) (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 role. (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 のサービスなどの信頼されたエンティティは、ロールを引き継ぎ、そのアクセス許可を継承できます。Amazon GameLift FleetIQ を使用するときには、Amazon GameLift FleetIQ が、AWS アカウントの Auto Scaling グループと EC2 インスタンスリソースを作成し、アクセスを許可するマネージド ポリシーを使用して IAM ロールを指定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、これらのリソースに対する直接的な権限を保持します。

- 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 で対応する Auto Scaling グループを使用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構成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 アカウントに新しい 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 グループ内のインスタンスでホストされ、Amazon GameLift FleetIQ に登録されているメンバーのゲームサーバーを機能別のグループに分ける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 に登録されるときに作成され、それが属するゲームサーバーグループを識別します。Amazon GameLift FleetIQ は、登録されたメンバーの各

ゲームサーバーの使用状況ステータスとクレームステータスを追跡します。これにより、ゲームサーバーの可用性のモニタリングを有効にします。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 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。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 1 対 1 の関係があります。Auto Scaling グループのすべての設定は更新できますが、Amazon GameLift FleetIQ はそのロジックのパートとして特定の設定を定期的に上書きして更新し、ゲームホスティングに利用できるようにスポットインスタンスを調整します。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。Amazon GameLift FleetIQ でのインスタンスは 1 つ以上のゲームサーバープロセスを実行します。各プロセスは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。

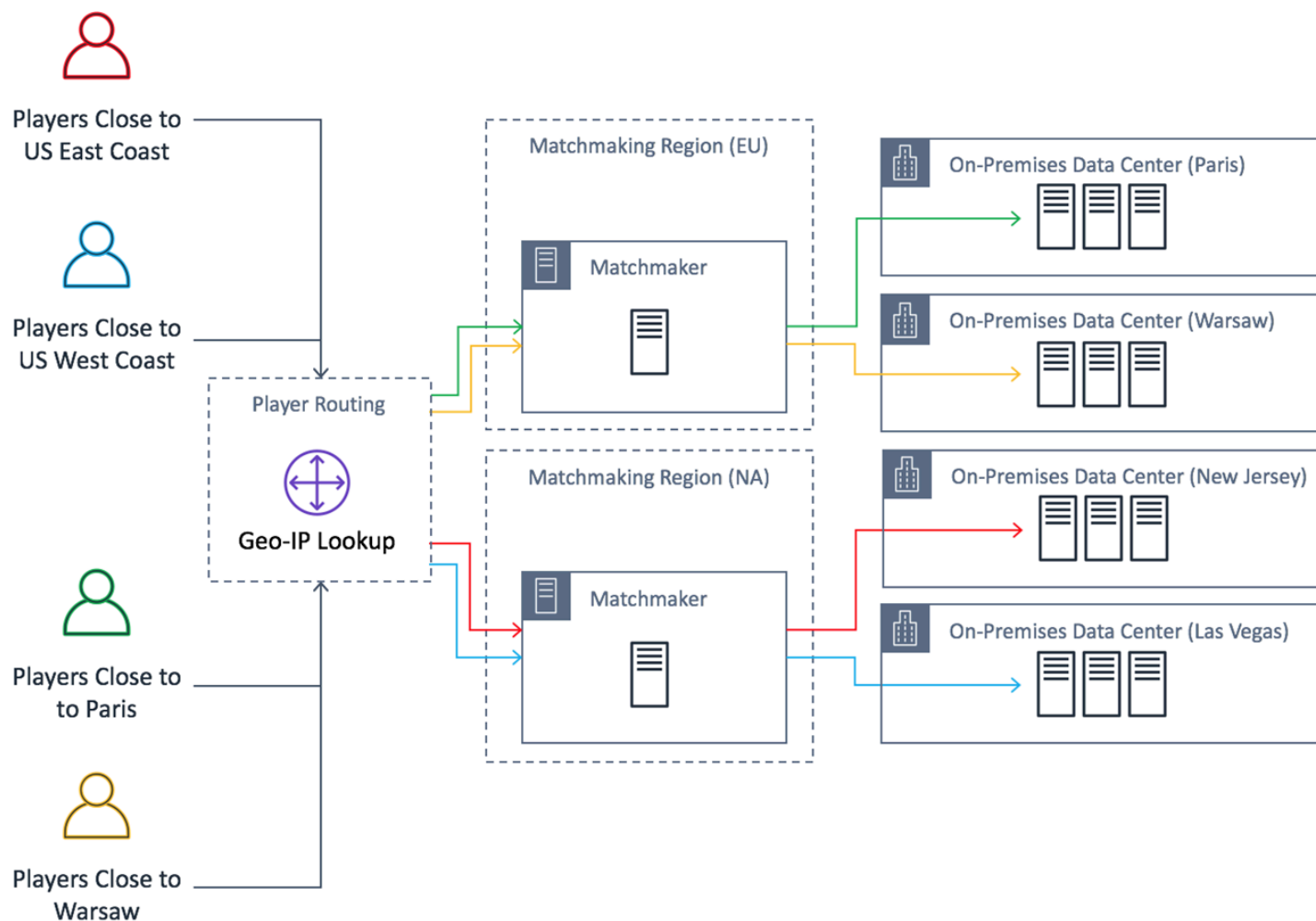
Amazon GameLift FleetIQ のゲームアーキテクチャ

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

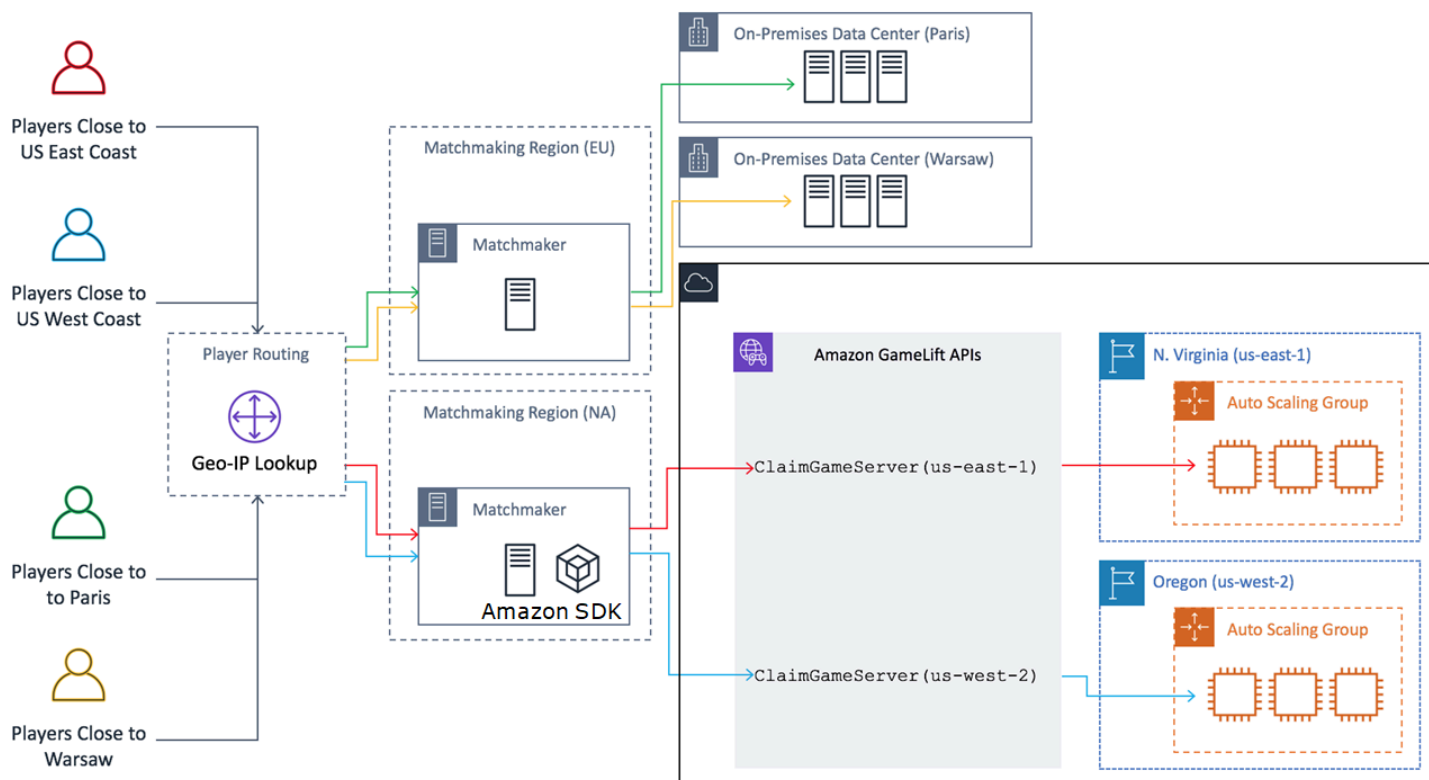
この例では、北米と欧州のプレイヤーをホストするための4つの専用データセンターで、ゲームホスティングが最初に処理されます。おおよその物理的な位置に応じて、プレイヤーは2つのリージョンマッチメーカーのいずれかにルーティングされます。マッチメーカーは、プレイヤーをスキルとレイテンシーに基づいてグループ化し、ラグを最小限に抑えるために近くのゲームサーバーに配置します。



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 の準備ができれば、デベロッパーは Amazon GameLift FleetIQ ゲーム サーバーグループを各 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。



Amazon GameLift FleetIQ のライフサイクルと状態

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、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。

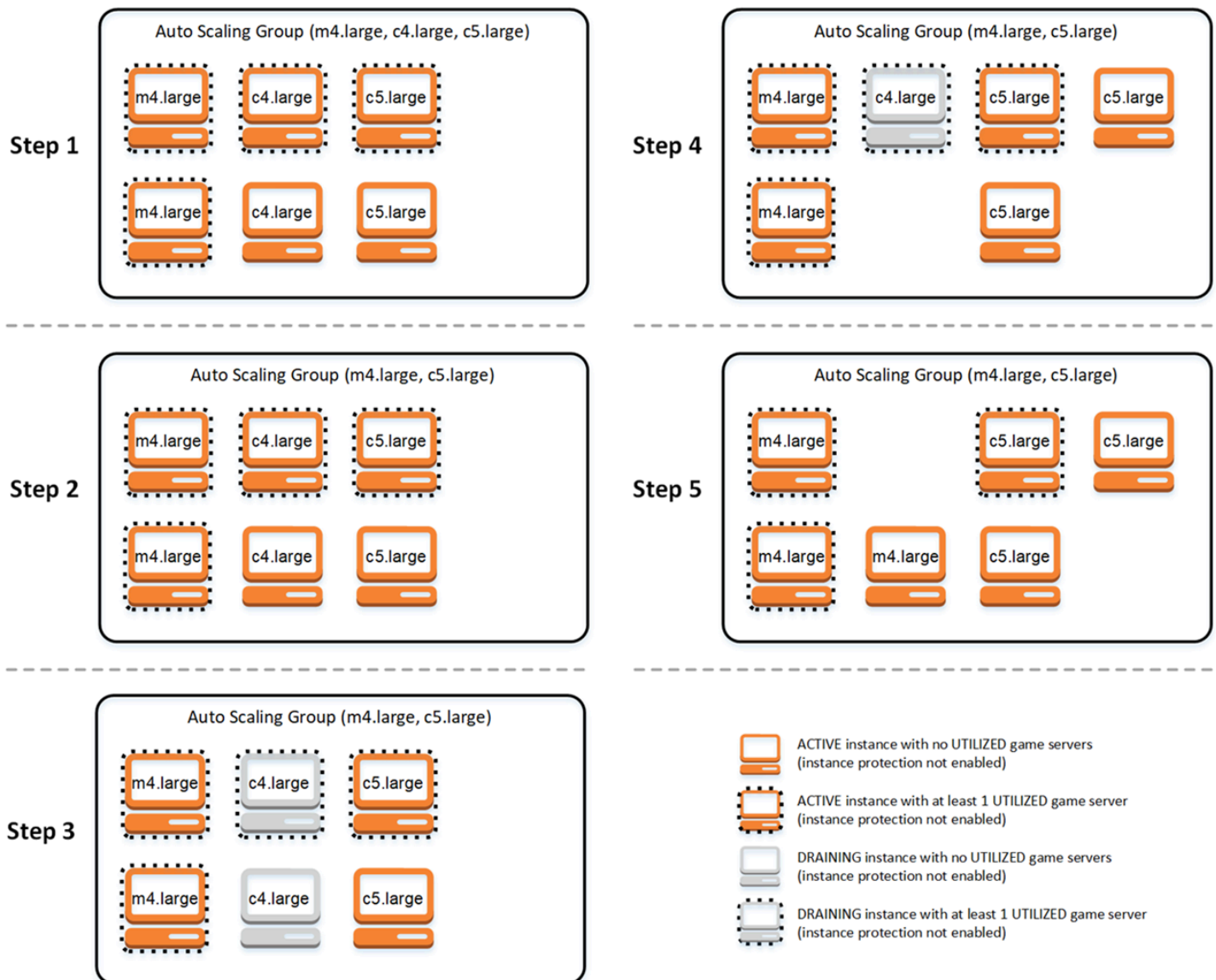
スポット再調整には、次の 2 つのキー目標があります。

- 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を引き起こす可能性が高いインスタンスを削除することによって再調整を行います。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。



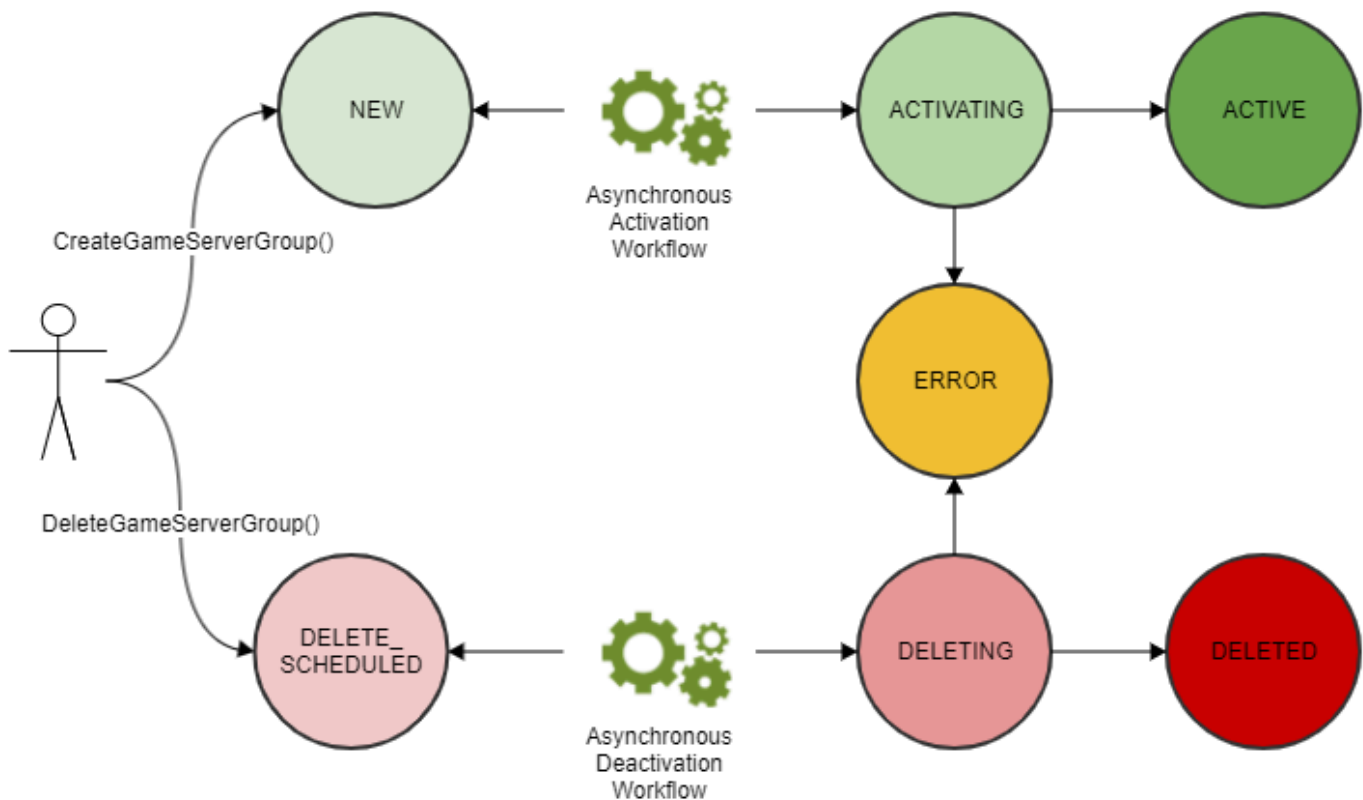
- [Step 1.](ステップ 1.) ゲームサーバーグループを介して、リンクされた Auto Scaling グループは、ゲームサーバーの保護が有効になっている m4.ラージ、c4.ラージ、c5.ラージ の各タイプのインスタンスを起動するように設定されます。Auto Scaling グループは、各タイプの 2 つのスポットインスタンスで構成されるバランスの取れたコレクションを起動しました。4 つのインスタンスには UTILIZED 状態 (破線の境界線で表示) のゲームサーバーが少なくとも 1 つあり、2 つのインスタンスはゲームプレイを現在サポートしていません。
- [Step 2.](ステップ 2.) Amazon GameLift FleetIQ は、3 つすべてのインスタンスタイプについて、現在のゲームホスティングに利用できるかどうかを評価します。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 グループ設定を即座に更新して、インスタンスタイプのリストから c4.ラージ を一時的に除外し、追加の c4.ラージインスタンスが起動されないようにします。

- ステップ 3。Amazon GameLift FleetIQ は、c4.ラージタイプの既存のインスタンスを識別し、これらをグループから除外するための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.](ステップ 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.](ステップ 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てきます。

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 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 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アクティベータリングに移行させます。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、Amazon GameLift FleetIQ に登録されたゲームサーバーを使用してインスタンスが起動されます。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を追跡し、ゲームサーバーに対する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 FleetIQ は、`DELETE_SCHEDULED` ステータスに応じて非同期の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` に移行させます。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

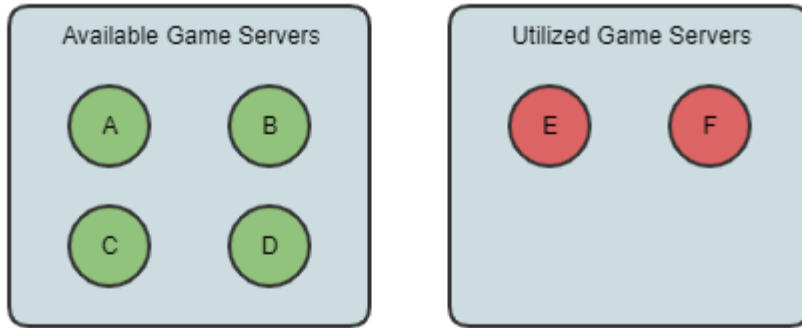
Amazon GameLift FleetIQ のゲームサーバーは、プロビジョニングやステータスの更新を含む、次のライフサイクルを経過します。ゲームサーバーは寿命が短いリソースと想定されます。ベストプラクティスとして、ゲームサーバーは、ゲームセッションの終了後に他のゲームセッションに再利用せずにメンバーを解除する必要があります。これにより、使用可能なゲームサーバーは、ゲームホスティングのために最低コストで利用できるリソースで常に実行されることを確実にします。

- ゲームサーバーリソースが作成されるのは、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 Amazon GameLift 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整っていることを Amazon GameLift FleetIQ に通知した時点です。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 FleetIQ は同じゲームサーバーを複数のリクエストに提供するこ

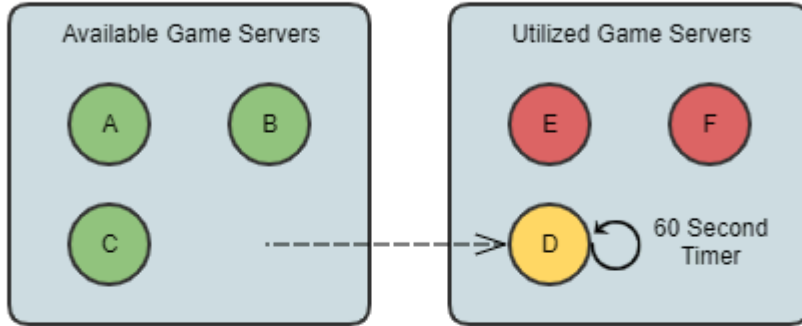
とを禁じられます。クレームステータスが空白になっているゲームサーバーは、クレームすることができません。

- 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。

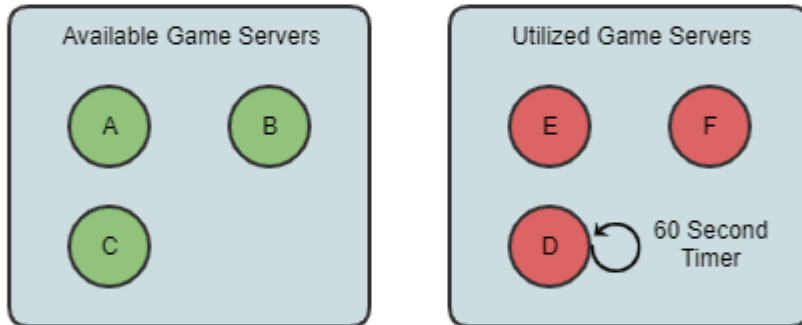
Step 1



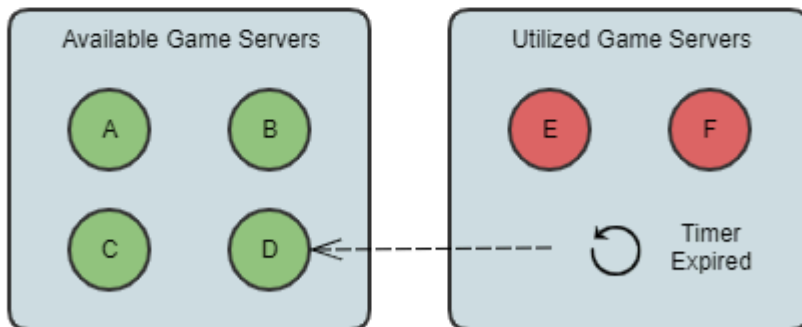
Step 2





Step 3a



Step 3b



-  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 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。このリクエストに応じて、Amazon GameLift FleetIQ は使用可能なゲームサーバー (D) を探し、そのクレームステータスを 60 秒間 CLAIMED に設定します。Amazon GameLift FleetIQ は、ゲームサーバーの接続情報 (IP アドレスとポート) およびその他のオプションのゲーム固有のデータのリクエストに対して反応します。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.](ステップ 3。) 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 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.](ステップ 3。) 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。シャットダウンの前に、ゲームサーバープロセスは Amazon GameLift API `DeregisterGameServer()` を呼び出して、ゲームサーバーのゲームサーバーグループのプールからの離脱を Amazon GameLift FleetIQ に通知します。

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ド機能 (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、バージョン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()` を使用します。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。
- 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。

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 の特徴

- [Optimized Spot balancing](最適化されたスポットバランシング) Amazon GameLift FleetIQは、定期的にインスタンスタイプを評価し、ゲームセッションの中断の可能性があるため、実行不可とみなされた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 FleetIQ APIs は、中断される可能性が低く対障害性に優れたスポットインスタンスに新しいゲームセッションをルーティングします。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。
- [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、ゲームサーバーの使用状況データを 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セットアップし、EC2 Auto Scaling グループ設定を維持し、他の AWS サービスと統合することができます。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。これらの設定を一時的に上書きしたり、必要に応じて Amazon GameLift FleetIQ のアクティビティを一時停止することができます。
- [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、Linux、コンテナ、Kubernetes など、現在 Amazon EC2 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。Amazon GameLift FleetIQ を使用すると、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。(ご利用いただけるかどうかは 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

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。

- Worldwide reach(ワールドワイドリーチ) Amazon GameLift FleetIQ は、現在、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。詳細なリストについては、AWS 全般のリファレンスの「[Amazon GameLift エンドポイントとクォータ](#)」を参照してください。

Amazon GameLift FleetIQ の料金

Amazon GameLift では、インスタンスについては使用時間ごとに課金され、帯域幅については転送されたデータ量によって課金されます。Amazon GameLift の課金および料金の詳細な一覧については、「[Amazon GameLift の料金表](#)」を参照してください。

Amazon GameLift でのゲームのホスティングまたはマッチメイキングのコストの計算については、[AWS Pricing Calculator](#) の使用方法について説明している「[Amazon GameLift 価格見積もりの生成](#)」を参照してください。

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 対応のソフトウェア](#)
- [Amazon GameLift FleetIQ のAWSアカウントを設定する](#)

Amazon GameLift FleetIQ 対応のソフトウェア

Amazon GameLift FleetIQ は、Amazon EC2 でホストする 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、およびゲームサービスをデプロイするために使用します。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

Amazon GameLift FleetIQ は、EC2 でサポートされている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。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

もしゲームサーバーがコンテナを使用している場合、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ています。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では、Amazon GameLift FleetIQ サービスと通信するための何らかの統合が必要です。ゲームは、AWS SDK に API コールを実行します。[AWS SDK をダウンロードするか、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 をサポートする AWS SDK は以下の言語で利用可能です。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 EC2、Auto Scaling、およびその他の サービスで Amazon GameLift FleetIQ を使用するには、必要なアクセス許可AWS アカウントを持つ を設定する必要があります。Auto Scaling AWS 以下のタスクを実行します。

- Amazon GameLift FleetIQ で使用するAWSアカウントをまだお持ちでない場合は、新しいアカウントを作成します。[AWS アカウントの作成](#) 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を管理する](#) を参照してください。
- IAM ロールを作成して、Amazon GameLift と Amazon EC2 リソースがやり取りできるようにします。[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。

AWS アカウント の作成

Amazon GameLift FleetIQ 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。AWS アカウント の作成には料金はかかりません。

トピック

- [AWS アカウントへのサインアップ](#)
- [管理ユーザーの作成](#)

AWS アカウントへのサインアップ

AWS アカウントがない場合は、以下のステップを実行して作成します。

AWS アカウント にサインアップするには

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、通話呼び出しを受け取り、電話のキーパッドを使用して検証コードを入力するように求められます。

AWS アカウントにサインアップすると、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、[管理ユーザーに管理アクセスを割り当て、ルートユーザーアクセスが必要なタスク](#)を実行する場合にのみ、ルートユーザーを使用してください。

サインアップ処理が完了すると、AWS からユーザーに確認メールが送信されます。<https://aws.amazon.com/> のアカウント 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。

管理ユーザーの作成

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

AWS アカウントのルートユーザーをセキュリティで保護する

1. ルートユーザー を選択し、AWS アカウント のメールアドレスを入力して、アカウント所有者として [AWS Management Console](#) 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 User Guide」の「[Signing in as the root user](#)」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 アカウントのルートユーザーの仮想 MFA デバイスを有効にする \(コンソール\)](#)」を参照してください。

管理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、管理ユーザーに管理アクセス権を付与します。

IAM アイデンティティセンターディレクトリ をアイデンティティソースとして使用するチュートリアルについては、「AWS IAM Identity Center ユーザーガイド」の「[IAM アイデンティティセンターディレクトリ デフォルトでのユーザーアクセスの設定](#)」を参照してください。

管理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 アイデンティティセンターの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](#)」を参照してください。

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

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と関連する Amazon EC2 および Auto Scaling サービスを使用するユーザーは、これらのサービスにアクセスするためのアクセス許可を持っている必要があります。

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。

次の手順に従って、AWS アカウント内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](#)
- [AWS CloudFormation で使用する追加のアクセス許可構文](#)

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。

- AWS IAM Identity Center のユーザーとグループ:

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「[アクセス許可一式を作成](#)」の手順を実行します。

- 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 \(フェデレーション\) 用のロールの作成](#)」を参照してください。

- IAM ユーザー:

- ユーザーが継承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 ユーザー用ロールの作成](#)」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 \(コンソール\) へのアクセス許可の追加](#)」の指示に従います。

Amazon GameLift FleetIQ _policy

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Action":
      [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition":
      {
        "StringEquals":
        {
          "iam:PassedToService": "gamelift.amazonaws.com"
        }
      }
    },
    {
      "Action":
```

```
[
  "iam:CreateServiceLinkedRole"
],
"Effect": "Allow",
"Resource": "arn:*:iam:*:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling"
},
{
  "Action":
  [
    "autoscaling:CreateAutoScalingGroup",
    "autoscaling:CreateOrUpdateTags",
    "autoscaling:DescribeAutoScalingGroups",
    "autoscaling:ExitStandby",
    "autoscaling:PutLifecycleHook",
    "autoscaling:PutScalingPolicy",
    "autoscaling:ResumeProcesses",
    "autoscaling:SetInstanceProtection",
    "autoscaling:UpdateAutoScalingGroup",
    "autoscaling>DeleteAutoScalingGroup"
  ],
  "Effect": "Allow",
  "Resource": "*"
},
{
  "Action":
  [
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeSubnets",
    "ec2:RunInstances",
    "ec2:CreateTags"
  ],
  "Effect": "Allow",
  "Resource": "*"
},
{
  "Action":
  [
    "events:PutRule",
    "events:PutTargets"
  ],
  "Effect": "Allow",
  "Resource": "*"
}
```

```

]
}

```

追加のAWS CloudFormation アクセス権限

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には、ポリシー構文に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

AWS Management Console の外部で AWS を操作するには、プログラマチックアクセス権が必要です。プログラマチックアクセス権を付与する方法は、AWS にアクセスしているユーザーのタイプによって異なります。

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 Identity Center 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 SDK、または AWS API へのプログラマチックリクエストに署名します。	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。 <ul style="list-style-type: none"> AWS CLI については、AWS Command Line Interface ユーザーガイドの「AWS IAM Identity Center を使用するための AWS CLI の設定」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"> • AWS SDK、ツール、および AWS API については、AWS SDK とツールリファレンスガイドの「IAM Identity Center 認証」を参照してください。
IAM	一時的な認証情報を使用して、AWS CLI、AWS SDK、または AWS API へのプログラムによるリクエストに署名します。	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 」の指示に従ってください。
IAM	(非推奨) 長期的な認証情報を使用して、AWS CLI、AWS SDK、AWS API へのプログラムによるリクエストに署名します。	<p>使用するインターフェイス用の手順に従ってください。</p> <ul style="list-style-type: none"> • AWS CLI については、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 SDK とツールについては、AWS 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て認証する」を参照してください。 • AWS API については、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 アクセスキー管理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 FleetIQ が Amazon EC2 インスタンスと Auto Scaling グループと連携するには、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。

- [Amazon GameLift FleetIQ のロールを作成する](#) で Amazon EC2 リソースを更新します。
- [Amazon EC2 のロールを作成する](#) 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管理ポリシーをアタッチします。[IAM ロールとマネージド ポリシーの詳細については、「AWS のサービス用のロールの作成」](#)および「[AWS マネージド ポリシー](#)」を参照してください。

Console

以下のステップでは、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 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. ユースケースの選択で、サービスのGameLiftリストから選択します。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。

- リスト「アタッチされたアクセス許可ポリシー」には、「」というポリシーが1つ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。

- [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。
- [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。
 - [Trusted entities](信頼されたエンティティ)は「AWS service: gamelift.amazonaws.com」に設定されています。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。

- 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 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 を使用して Amazon GameLift AWS 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。

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 FleetIQ ゲームサーバーグループを設定するときに必要になります。

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。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 [ゲームリフト create-game-server-group](#) を呼び出します。が ACTIVE 状態 GameServerGroup になると、Amazon GameLift FleetIQ 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 Amazon GameLift FleetIQ と通信できるようになります。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、ヘルスステータスをレポートする必要があります。Amazon GameLift FleetIQ ゲームサーバーグループを作成するときに、Amazon EC2 起動テンプレートを使用して IAM インスタンスプロファイルにこのロールを含めます。

AWSCLI を使用して、Amazon EC2 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 サービス用のロールの作成](#)」を参照してください。

AWS CLI

以下の手順では、を使用して Amazon EC2 のカスタム Amazon アクセス GameLift 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、新しいロールの名前で [iam list-role-policies](#) を呼び出します。

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

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 サーバーグループを管理する](#)
- [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、「[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 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 サーバーグループを管理する](#)」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 サーバーグループを管理する

このトピックでは、Amazon GameLift FleetIQ ゲーム サーバーグループを設定するため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。
- GameLift FleetIQ が Auto Scaling グループを作成し、相互作用することを許可するために AWS アカウントへの制限されたアクセスを拡張する 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 グループを作成します。ゲームサーバグループを作成するときは、Amazon GameLift FleetIQ のゲーム固有の設定を指定します。これには、balancing戦略やインスタンスタイプの定義が含まれます。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 サーバーグループを更新する

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。これには、リソースタイプの最適化も含まれます。プロパティを更新するには、[\[UpdateGameServerGroup\]](#) (更新ゲームサーバーグループ) を呼び出します。ゲーム サーバーグループへの変更が有効になると、Amazon GameLift FleetIQ が Auto Scaling グループ内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

デプロイされたインスタンスでゲームサーバーグループと 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。

[\[Amazon GameLift コンソール\]](#)の [ゲームサーバーグループ] を使用して、ゲームサーバーグループのステータスをモニタリング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](#)」の追加ガイド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から、[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 がサーバーキャパシティに関する情報を要求したり、ゲームサーバーをクレームしたりするときに、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 응답できます。登録時、ゲームサーバーは、受信クライアント接続に使用するポートや 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 Amazon GameLift FleetIQ に提供できません。

```
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メンバーになると、Amazon GameLift FleetIQ でサーバーキャパシティの状態を同期させるために、定期的にヘルスおよび使用状況を報告する必要があります。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、このメカニズムを使用して Amazon GameLift FleetIQ のゲームサーバーのヘルスアップデートをトリガーできます。

[Utilization status] (使用率ステータス)

ゲームサーバーの使用状況を報告すると、Amazon GameLift FleetIQ はどのゲームサーバーが現在理想的で、新しいゲームセッションに利用できるかを把握します。ゲームサーバーには、Amazon

GameLift FleetIQ の使用状況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、「[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。使用状況が更新されない場合、Amazon GameLift FleetIQ はクレームを削除し、ゲームサーバーが利用可能であると仮定し、別のクライアント要求のためにゲームサーバーを予約することができます。

```
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 AAmazon GameLift FleetIQ から登録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 が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を使用し、自動スケールリングを行います。

- 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)。

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

Amazon GameLift FleetIQ に使用可能なゲームサーバーを選択させるには、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、ゲームホスティング用に有効で、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出す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への応答で、Amazon GameLift FleetIQ は、クライアントがゲームサーバーに接続するために使用できる GameServer リソース、接続情報、ゲームデータを特定します。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スタート後に、ゲームサーバーまたはクライアントサービスが Amazon GameLift FleetIQ 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。これは、Amazon GameLift FleetIQ が、ゲームサーバーキャパシティに対する後続の要求に応じてこのゲームサーバーを提供しないようにするためのものです。 [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 FleetIQ は、[Amazon GameLift FleetIQ ロジック](#) に説明されているようなインスタンスタイプの最適化ロジックを実行しません。

```
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 リファレンスガイド

このセクションには、Amazon GameLift FleetIQ を使用するためのリファレンスドキュメントが含まれます。

トピック

- [Amazon GameLift FleetIQ サービス API リファレンス \(AWS SDK\)](#)

Amazon GameLift FleetIQ 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示します。Amazon GameLift FleetIQ サービス API は `aws.gamelift` 名前空間の AWS SDK 内にあります。[AWS SDK をダウンロードする](#)か、[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 EC2 のクラウドベースのゲームホスティング向け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。Amazon GameLift のホスティングオプションの詳細については、[Amazon GameLift 開発者ガイド](#)を参照してください。

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 グループと組み合わせて管理できます。

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 インスタンスのゲームサーバープロセスの起動方法、セットアップおよび Auto Scaling グループを制御し、FleetIQ 最適化の適用方法を定義します。

- [CreateGameServerGroup](#) – 新しいゲームサーバーグループおよび対応する 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:[ゲームサーバーグループの作成](#)
- [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:[ゲームサーバーグループのリスト化](#)
- [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:[ゲームサーバーグループの説明](#)

- [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:[ゲームサーバーグループの更新](#)
- [DeleteGameServerGroup](#) – ゲームサーバーグループを完全に削除し、関連付けられてい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:[ゲームサーバーグループの削除](#)
- [ResumeGameServerGroup](#) – ゲームサーバーグループで中断された FleetIQ アクティビティを復元します。CLI コマンド:[ゲームサーバーグループの再開](#)
- [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:[ゲームサーバーグループの停止](#)

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 インスタンスのゲームサーバープロセスの起動方法、セットアップおよび Auto Scaling グループを制御し、FleetIQ 最適化の適用方法を定義します。

- [RegisterGameServer](#) – 新しいゲームサーバーから呼び出して、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:[ゲームサーバーグループの登録](#)
- [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:[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ように Amazon GameLift FleetIQ に指示します。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

Amazon GameLift をサポートする AWS SDK は以下の言語で利用可能です。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 を使用したセキュリティ

Amazon GameLift FleetIQ を Amazon EC2 のスタンドアロン機能として使用している場合は、Linux インスタンス用 Amazon EC2 ユーザーガイドの「[Amazon EC2 におけるセキュリティ](#)」を参照してください。

AWS では、クラウドセキュリティを最優先事項としています。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 とユーザーの間の責任共有です。Amazon GameLift FleetIQ を使用する際に責任共有モデルを適用する方法の詳細については、「[Amazon GameLift のセキュリティ](#)」を参照してください。。

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、新しいFleetIQに関する特徴、更新、サービスに関する修正の詳細が記載されています。このページには Amazon GameLift SDK のバージョン履歴も含まれています。

Amazon GameLift デベロッパーリソース

すべての Amazon GameLift ドキュメントとデベロッパー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。