



このコンテンツは公開から3年以上経過しており内容が古い可能性があります
最新情報については[サービス別資料](#)もしくはサービスのドキュメントをご確認ください

[AWS Black Belt Online Seminar]

Amazon GameLift

サービスカットシリーズ

Solutions Architect 安藤 怜央
2019/10/09

AWS 公式 Webinar
<https://amzn.to/JPWebinar>



過去資料
<https://amzn.to/JPArchive>



自己紹介

名前

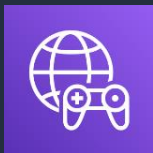
安藤 怜央 (あんどう れおう)

所属

アマゾン ウェブ サービス ジャパン株式会社
技術統括本部
ソリューション アーキテクト



好きな AWS のサービス



Amazon GameLift



AWS CodePipeline

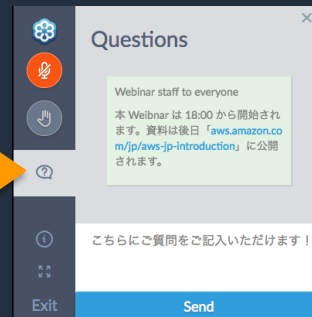
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾンウェブサービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
お答えできませんのでご了承ください

- ① 吹き出しをクリック
- ② 質問を入力
- ③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2019年10月9日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<http://aws.amazon.com>)にてご確認ください。
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。
- 価格は税抜表記となっております。日本居住者のお客様が東京リージョンを使用する場合、別途消費税をご請求させていただきます。
- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at <http://aws.amazon.com/agreement/>. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

本日のアジェンダ

- マルチプレイヤーゲームについて
- Amazon GameLift とは
- Amazon GameLift によるサーバーのホスティング
- Amazon GameLift ホスティングリソース
- Amazon GameLift の応用例
- 補足情報やご利用料金について
- まとめ

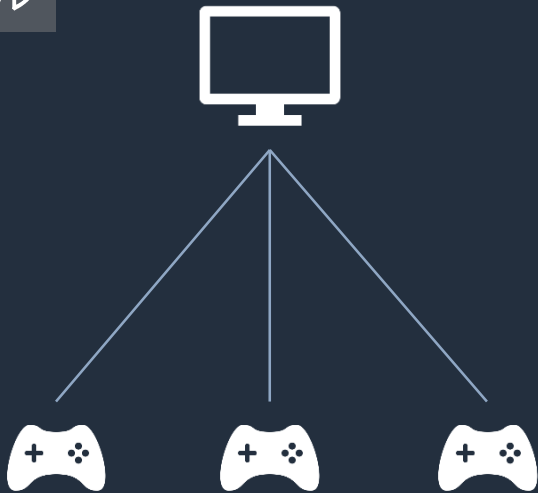
本日のアジェンダ

- マルチプレイヤーゲームについて
- Amazon GameLift とは
- Amazon GameLift によるサーバーのホスティング
- Amazon GameLift ホスティングリソース
- Amazon GameLift の応用例
- 補足情報やご利用料金について
- まとめ

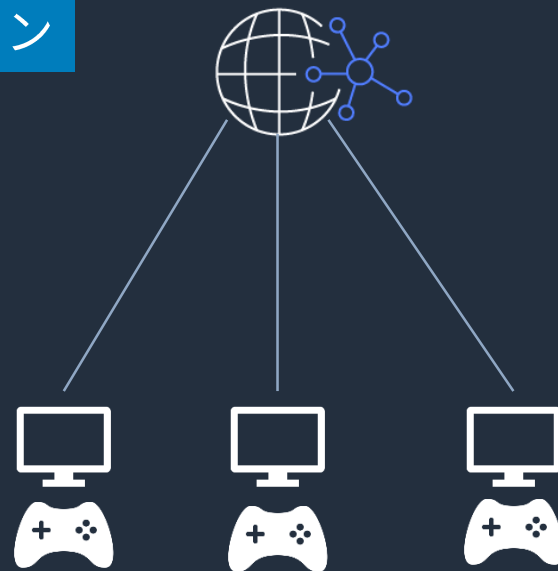
マルチプレイヤーゲームとは

他のプレイヤーと一緒に同じゲーム環境で対戦・協力して遊ぶゲームモードのこと

ローカル

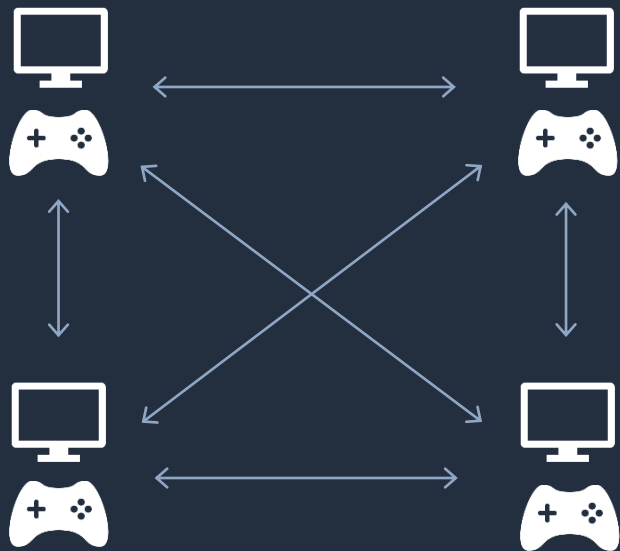


オンライン

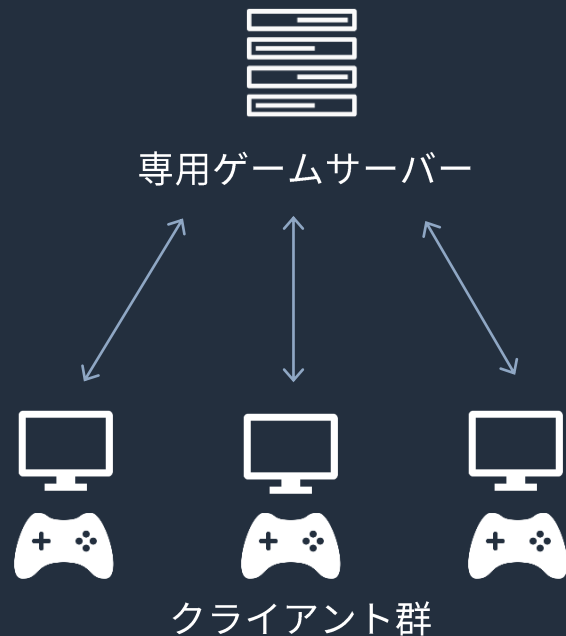


オンラインマルチプレイヤーゲームのネットワーク構造

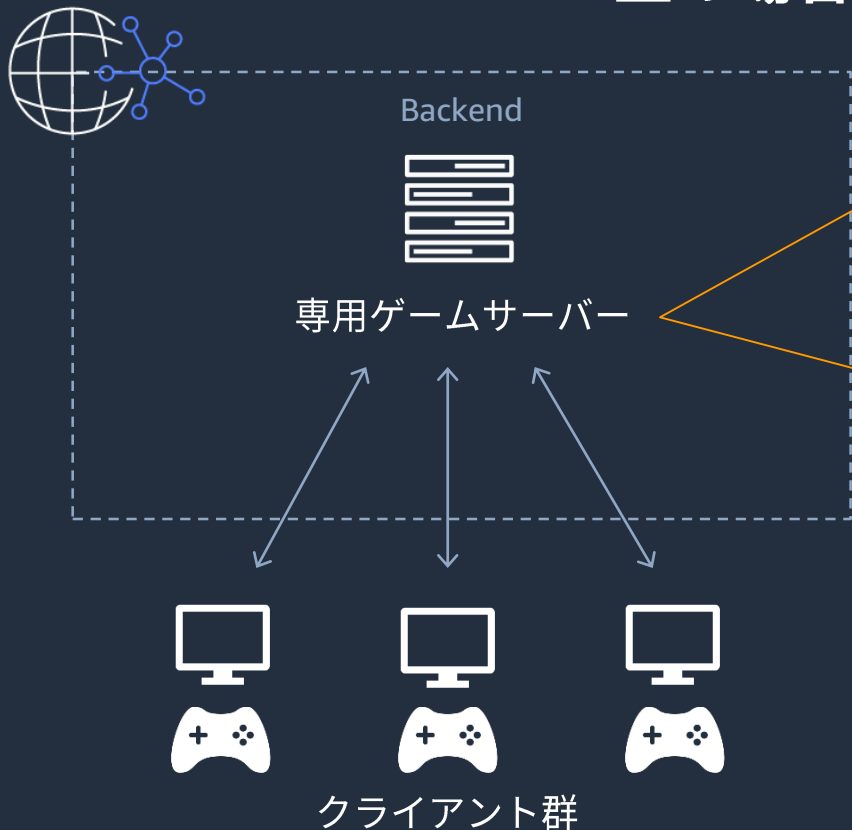
Peer to Peer (P2P)



Client/Server



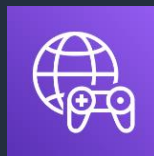
オンラインマルチプレイヤーゲームのネットワーク構造： Client/Server 型の場合



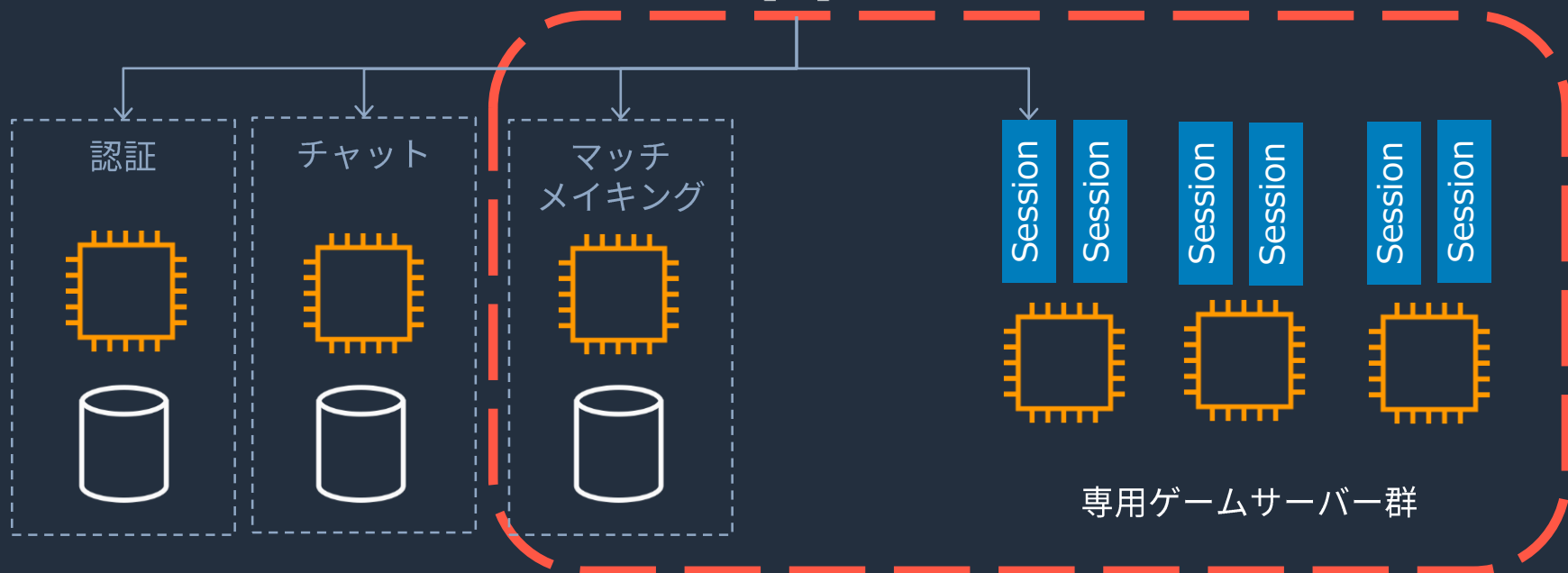
- プレイヤーをどのように接続させる？
- 通信方式には何を選択する？
- 開発のためにどんな専門的な知識が必要か？
- 低レイテンシーをどう実現する？
- スケーリングはどのように行う？
- 可用性を高める方法は？
- セキュリティはどう確保する？
- コストはどれぐらいかかる？

など.....

Client/Server 型マルチプレイヤーゲームのアーキテクチャ例



Amazon GameLift
カバー領域



本日のアジェンダ

- マルチプレイヤーゲームについて
- Amazon GameLift とは
- Amazon GameLift によるサーバーのホスティング
- Amazon GameLift ホスティングリソース
- Amazon GameLift の応用例
- 補足情報やご利用料金について
- まとめ

Amazon GameLift とは

クラウド内でセッションベースのマルチプレイヤー専用ゲームサーバーを
デプロイ、操作、スケーリングするマネージド型サービス



数百万のプレイヤーに対応できるよう
専用ゲームサーバーをスケーリング・ホスティング

AWS グローバルインフラストラクチャ上で稼働

DDoS 攻撃から保護するように設計

待機時間とレイテンシーを
最小に抑えたゲーム体験を実現

柔軟にカスタマイズできる
マッチメイキング機能を提供

Amazon GameLift の概念

Server Hosting



カスタムゲーム
サーバー



リアルタイム
サーバー

Hosting Resource



フリート



エイリアス



キュー



FlexMatch

本日のアジェンダ

- マルチプレイヤーゲームについて
- Amazon GameLift とは
- Amazon GameLift によるサーバーのホスティング
- Amazon GameLift ホスティングリソース
- Amazon GameLift の応用例
- 補足情報やご利用料金について
- まとめ

Amazon GameLift による専用ゲームサーバーのホスティング

Amazon GameLift は 2 種類の方法でサーバーをホスティング可能



カスタムゲームサーバー

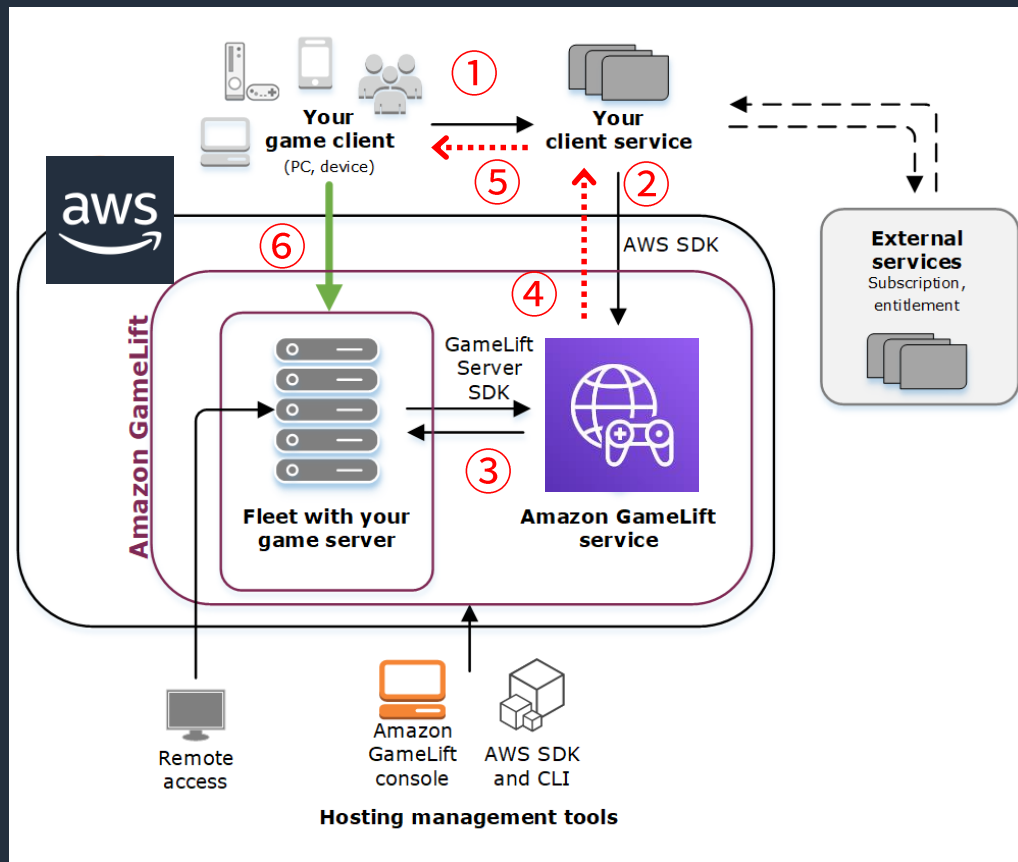


リアルタイムサーバー

カスタムゲームサーバーの主要コンポーネント

主要コンポーネント	役割
ゲームサーバー	クラウドで実行されるゲームのサーバーソフトウェア
ゲームセッション	プレイヤーの接続先となるゲームサーバーのインスタンス
Amazon GameLift サービス	ゲームサーバーをホストするためのリソースを管理するコアサービス
ゲームクライアント	デバイスで実行されているゲームのソフトウェア
クライアントサービス (ゲームサービス)	GameLift サービスとゲームクライアント間の通信を仲介するサービス 認証認可などゲーム固有のロジックも処理

カスタムゲームサーバーにおけるゲームのアーキテクチャ



カスタムゲームサーバーにおけるゲームへの統合

クライアントサービスとゲームサーバーで専用の SDK を組み込む
ゲームクライアントは自前の方法でゲームサーバーに接続



ゲームクライアント

- クライアントサービスから受け取った情報を使ってゲームサーバーに直接接続する



クライアントサービス

- AWS SDK を使って Amazon GameLift サービスとやりとり
- ゲームクライアントに必要な情報を引き渡す



ゲームサーバー

- Amazon GameLift Server SDK を使って Amazon GameLift サービスとやりとり

カスタムゲームサーバーの開発

主要なゲームエンジンを使用してカスタムゲームサーバーを実装し、OS (Linux, Windows) 向けにビルドを作成



Amazon GameLift Server SDK

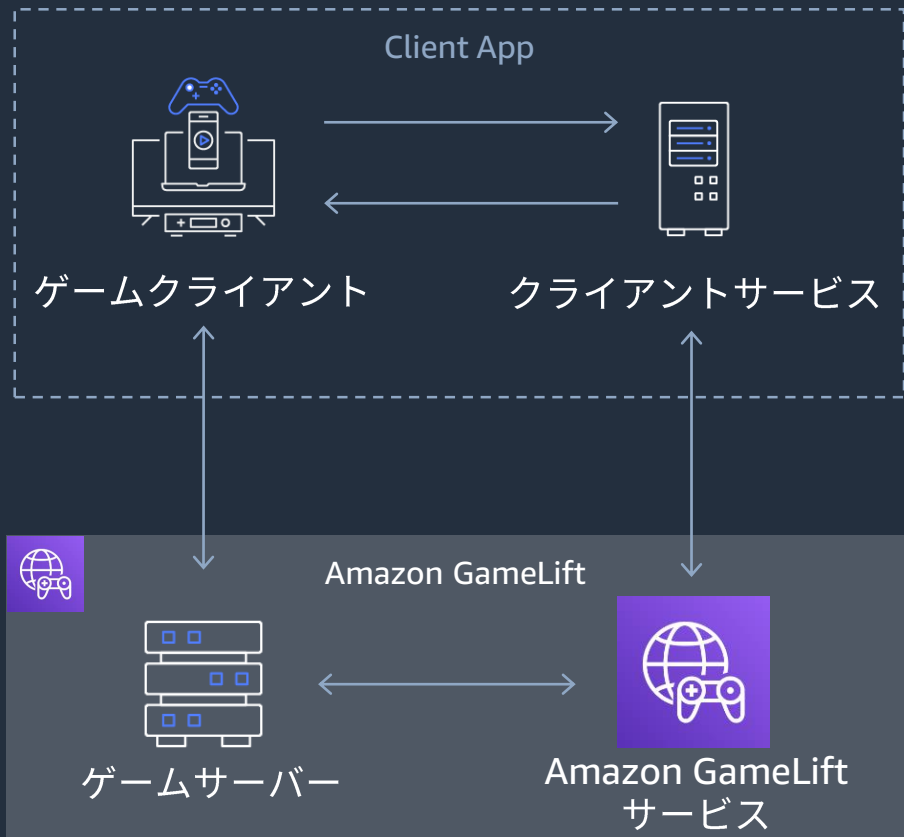
- C# (.NET)
- C++ for Unreal Engine
- C++



ゲームエンジン

- Unity
- Unreal Engine
- Amazon Lumberyard
- C++ または C# ライブラリをサポートするエンジン

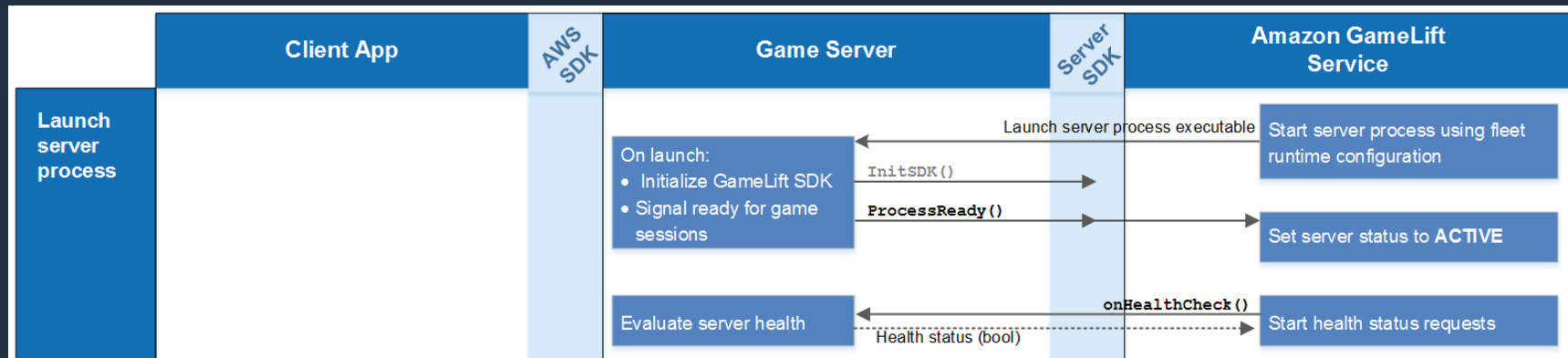
各コンポーネントのやりとり



- サーバープロセス起動時
- ヘルスチェック時
- ゲーム開始時
- プレイヤー追加
- プレイヤー離脱
- ゲーム停止
- サーバーのシャットダウン

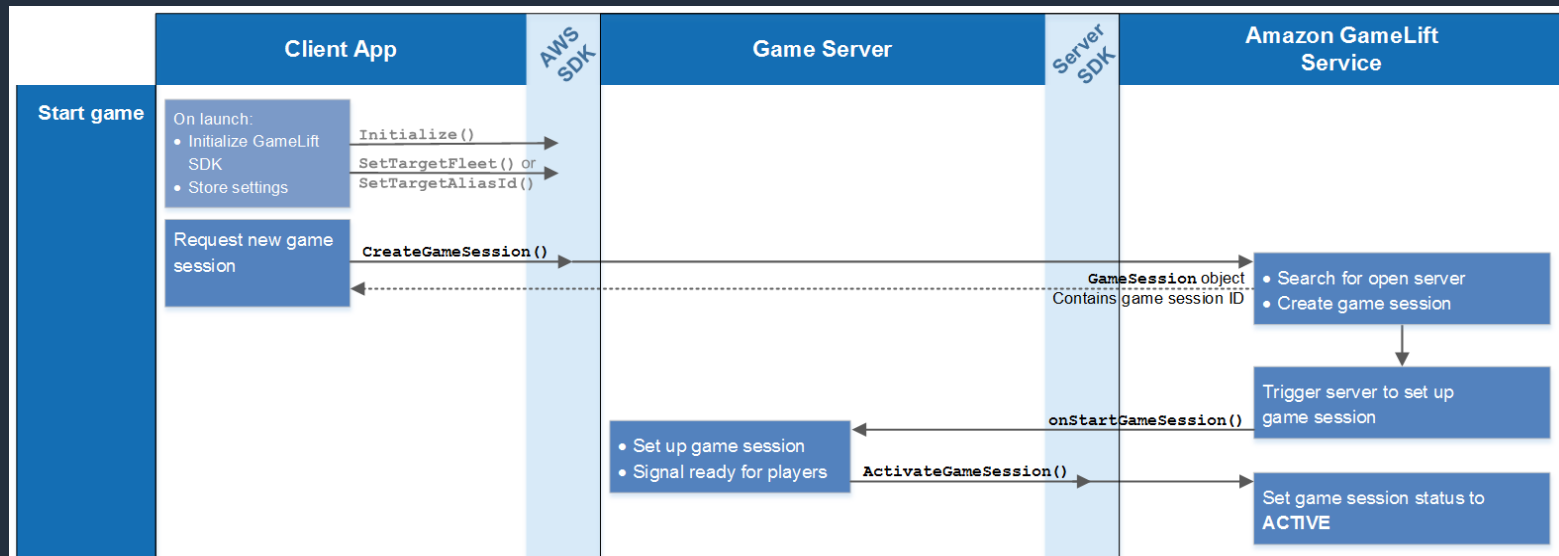
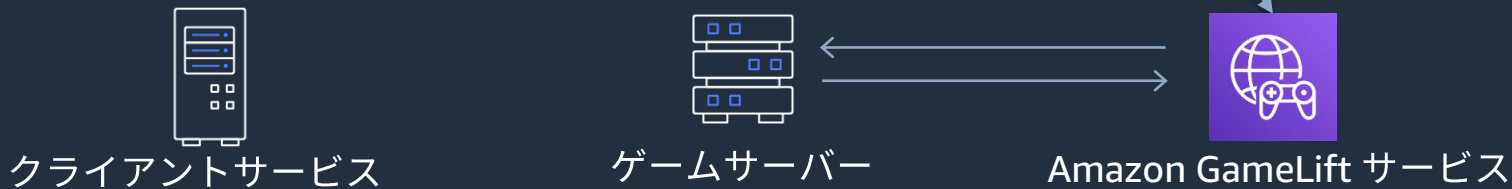
各コンポーネントのやりとり (1/4)

サーバープロセス起動時、ヘルスチェック時

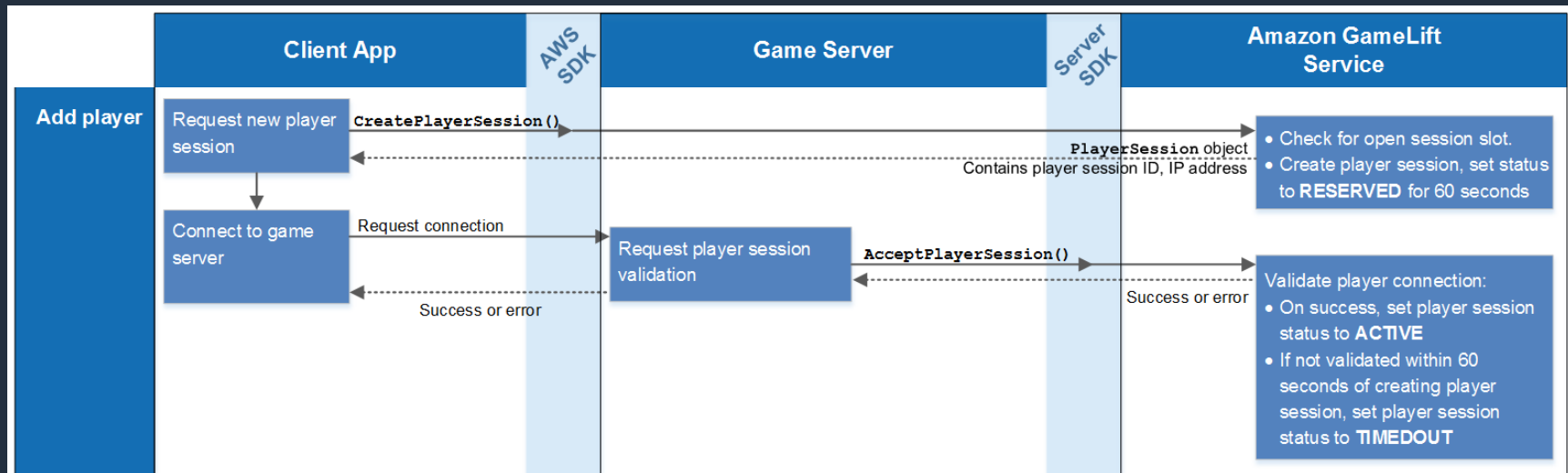
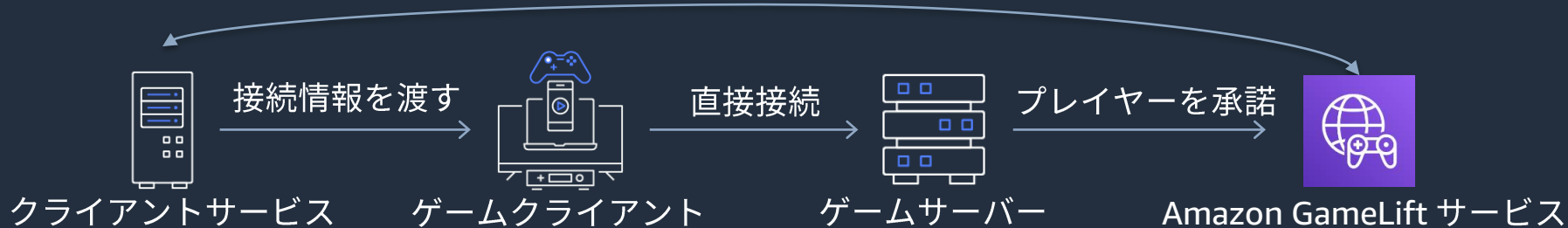


各コンポーネントのやりとり (2/4)

ゲーム開始時



各コンポーネントのやりとり (3/4) プレイヤー追加



各コンポーネントのやりとり (4/4)

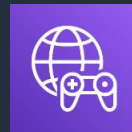
プレイヤー離脱, ゲーム停止, サーバーのシャットダウン



ゲームクライアント



ゲームサーバー

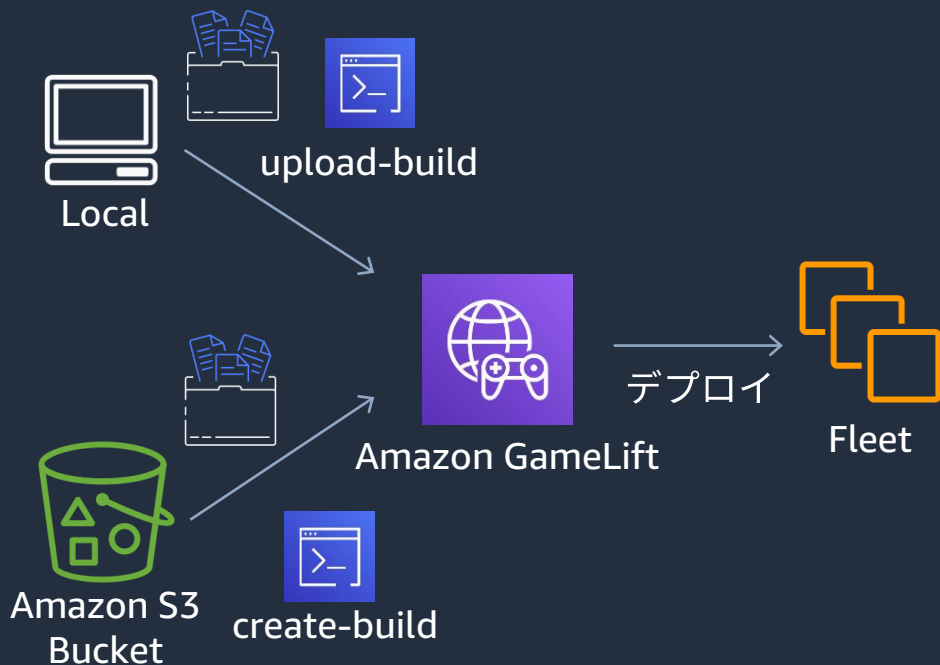


Amazon GameLift サービス

	Client App	AWS SDK	Game Server	Server SDK	Amazon GameLift Service
Drop player	Disconnect from server		Notify GameLift when lost connection is detected	<code>RemovePlayerSession()</code>	<ul style="list-style-type: none">Set player session status to COMPLETEDReopen player slot in game session
Stop game			Notify GameLift when game session ends	<code>TerminateGameSession()</code>	<ul style="list-style-type: none">Set game session status to TERMINATEDUpdate fleet utilization dataUpload game session logs
Shut down server process			Notify GameLift server process is terminating	<code>ProcessEnding()</code>	<ul style="list-style-type: none">Set server process status to TERMINATEDRecycle instance resources

カスタムゲームサーバービルドのアップロード・更新

AWS CLI 経由でローカルファイルか Amazon S3 内のファイルを指定ビルドを更新する場合も新規にフリートをデプロイ



- ビルドをアップロードすると毎回新しいビルドレコードとして登録
- 更新したビルドを指定するには
 - キューを使用する
 - エイリアスを使用する
 - 自動更新を組み込む

<https://aws.amazon.com/blogs/gametech/automating-deployments-to-amazon-gamelift/>

Amazon GameLift による専用ゲームサーバーのホスティング

Amazon GameLift は 2 種類の方法でサーバーをホスティング可能



カスタムゲームサーバー

- ゲームエンジンを使用してサーバーのビルドバイナリを開発
- パッケージ化したビルドをアップロードしフリートをデプロイ
- 詳細なゲームロジックを備え完全にカスタマイズさせた本格的なゲームサーバーとして最適



リアルタイムサーバー

NEW !!

Amazon GameLift Realtime Servers

game tech


Amazon GameLift リアルタイムサーバーとは

数行のスク립トでゲームサーバーを作成できる軽量サーバーソリューション
カスタムゲームサーバーと比較して複雑な処理を必要としないゲームに最適



Node.js ベースの JavaScript で実装

TCP, UDP によるメッセージング処理を提供

ステートレスとしてもステートフルとしても稼働

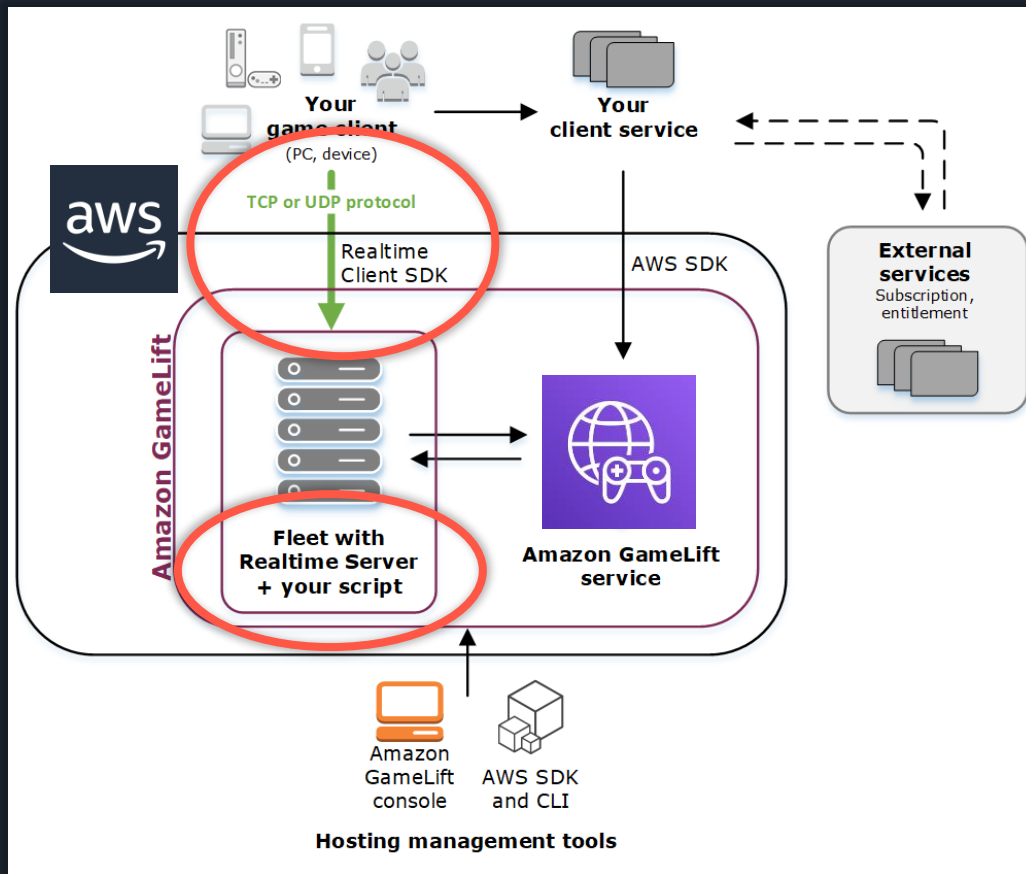
カスタムゲームサーバーと同様の
GameLift の機能を利用可能（一部を除く）

カードゲーム、ターンベースの戦略ゲーム、
軽量のモバイルゲームなどに最適

リアルタイムサーバーの主要コンポーネント

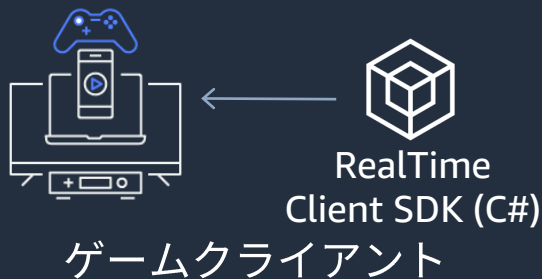
主要コンポーネント	役割
RealTime サーバー	RealTime スクリプトを実行するサーバーソフトウェア
ゲームセッション	プレイヤーの接続先となる RealTime サーバーのインスタンス
Amazon GameLift サービス	RealTime サーバーをホストするための リソースを管理するコアサービス
ゲームクライアント	デバイスで実行されているゲームのソフトウェア
クライアントサービス (ゲームサービス)	GameLift サービスとゲームクライアント間の 通信を仲介するサービス 認証認可などゲーム固有のロジックも処理

リアルタイムサーバーにおけるゲームのアーキテクチャ



リアルタイムサーバーにおけるゲームとの統合

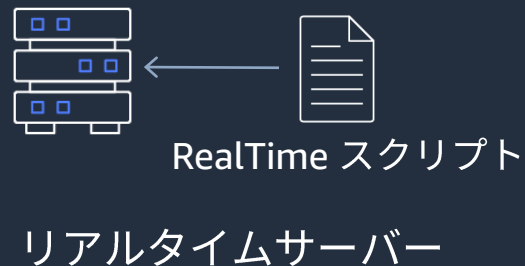
ゲームクライアントとクライアントサービスで専用の SDK を組み込む
リアルタイムサーバーはスクリプトを開発



- RealTime Client SDK (C#) を組み込み
クライアントサービスから受け取った情報を用いてリアルタイムサーバーに直接接続



- AWS SDK を使って Amazon GameLift サービスとやりとり
- ゲームクライアントに必要な情報を引き渡す



- RealTime スクリプトでコールバックやインターフェースを実装
- Amazon GameLift サービスとやりとり

リアルタイムサーバーの開発

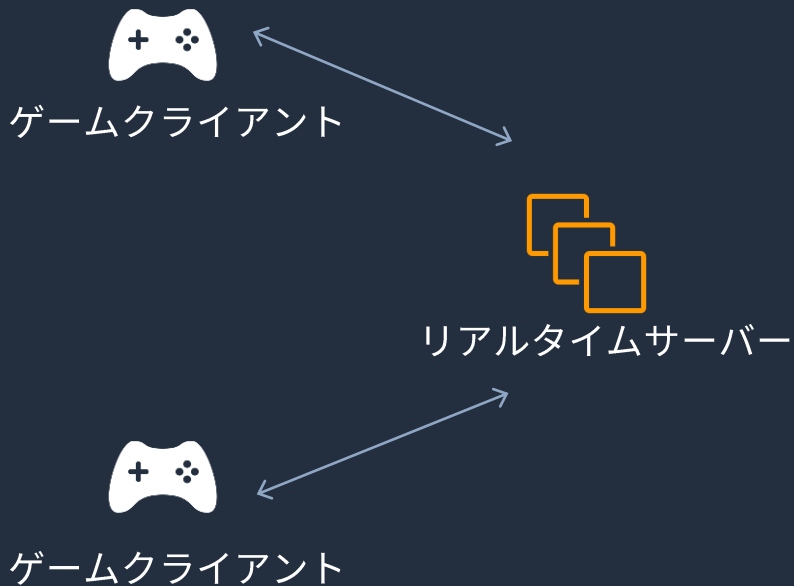
ゲームセッションのライフサイクル管理処理をスクリプトに記述
必要に応じてイベント駆動やメッセージ送信などのゲームロジックを追加

主なコールバック	タイミング
init	サーバーの初期化時（実装必須）
onMessage	サーバーにメッセージが送信されたとき
onHealthCheck	ゲームセッションの状態を尋ねられたとき
onStartGameSession	新しいゲームセッションが開始したとき
onPlayerAccepted	プレイヤーの接続が受け入れられたとき

https://docs.aws.amazon.com/ja_jp/gamelift/latest/developerguide/realtime-script-ref.html

クライアントとリアルタイムサーバーのやりとり

ゲームクライアント・サーバー間で SDK 経由のメッセージングを使用
メッセージはリアルタイムサーバーを介して送受信



通信プロトコル

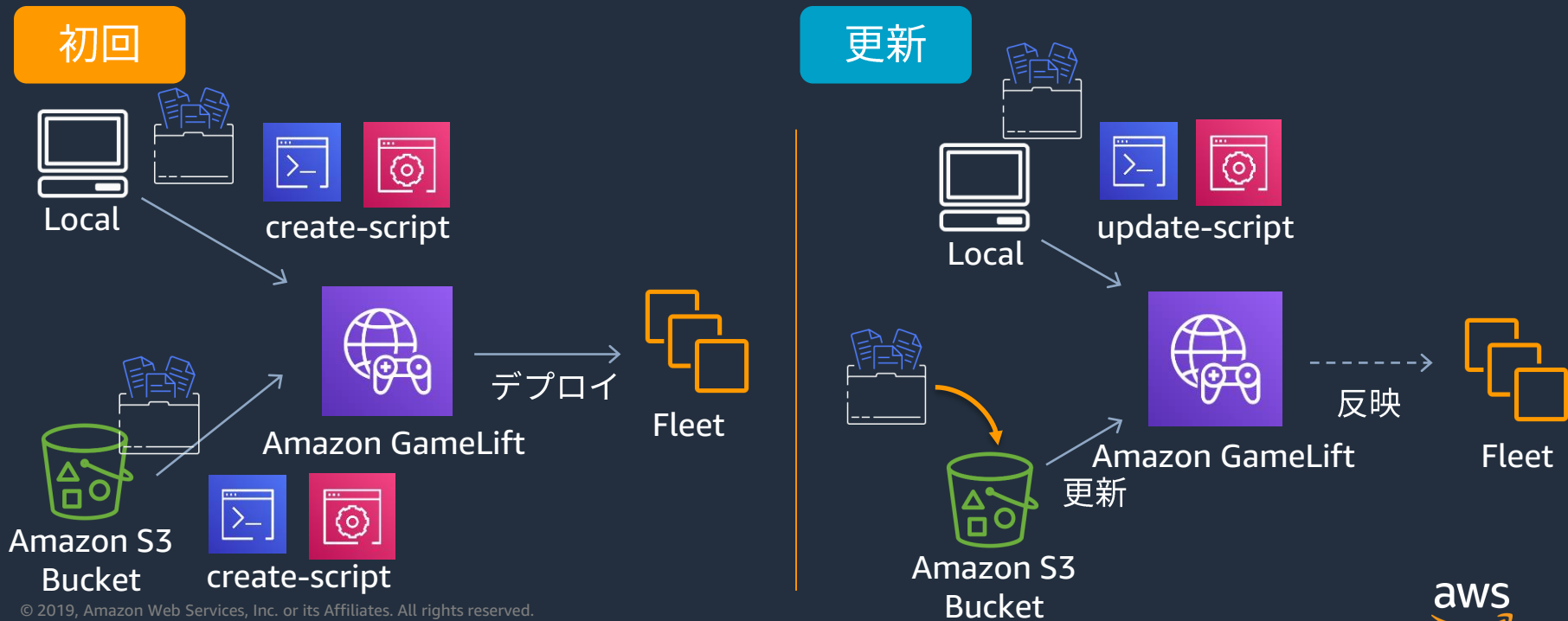
- 送信するメッセージ毎にどちらかを指定
 - 信頼性のある TCP 接続
 - 高速な UDP チャンネル

メッセージの内容

- オペレーションコードとペイロードで構成
- アクションとオペレーションコードのマッピングを開発者で定義して利用

RealTime スクリプトのアップロード・更新

AWS CLI またはコンソール上でローカルか Amazon S3 内のファイルを指定スクリプトを更新する場合にフリートのデプロイは不要



Amazon GameLift による専用ゲームサーバーのホスティング

Amazon GameLift は 2 種類の方法でサーバーをホスティング可能
ゲームのジャンルや仕様、スキルセットに合わせて選択



カスタムゲームサーバー

- ゲームエンジンを使用してサーバーのビルドバイナリを開発
- パッケージ化したビルドをアップロードしフリートをデプロイ
- 詳細なゲームロジックを備え完全にカスタマイズさせた本格的なゲームサーバーとして最適



リアルタイムサーバー

- Node.js ベースの JavaScript でサーバーのスクリプトを記述
- 記述したスクリプトをアップロードしフリートをデプロイ
- 複雑さを必要とせず簡単に迅速にゲームを起動させたい軽量のゲームサーバーとして最適

本日のアジェンダ

- マルチプレイヤーゲームについて
- Amazon GameLift とは
- Amazon GameLift によるサーバーのホスティング
- Amazon GameLift ホスティングリソース
- Amazon GameLift の応用例
- 補足情報やご利用料金について
- まとめ

Amazon GameLift の概念

Server Hosting



カスタムゲーム
サーバー



リアルタイム
サーバー

Hosting Resource



フリート



エイリアス



キュー



FlexMatch

Amazon GameLift の概念

Server Hosting



カスタムゲーム
サーバー



リアルタイム
サーバー

Hosting Resource



フリート



エイリアス



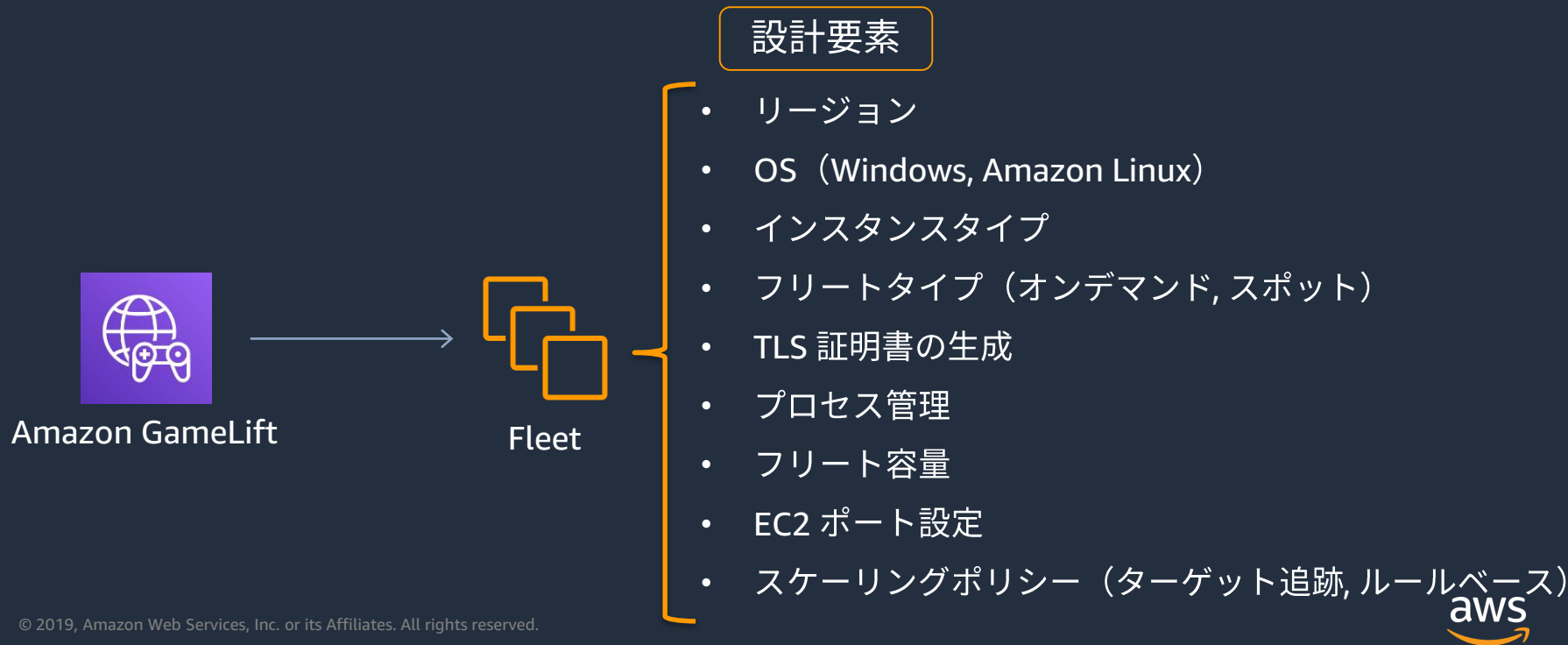
キュー



FlexMatch

フリートとは

EC2 インスタンス群形式のホスティングリソース
カスタムゲームサーバーまたはリアルタイムサーバーを用いてデプロイ



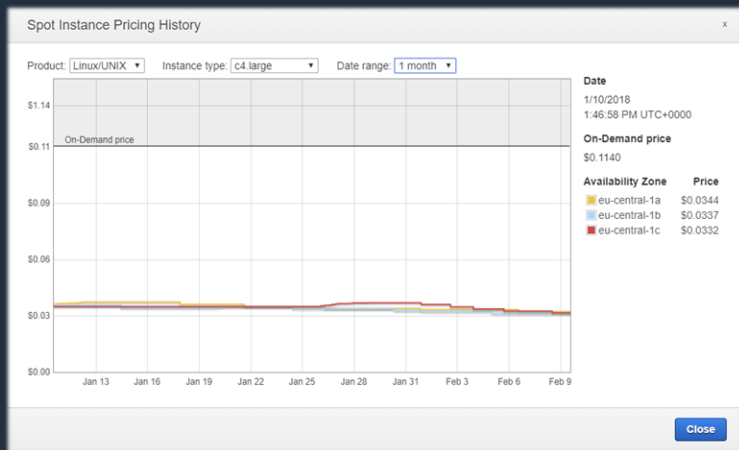
フリートでスポットインスタンスを活用

フリートタイプにスポットインスタンスを指定
最大で 90% の割引料金でゲームサーバーを稼働できる

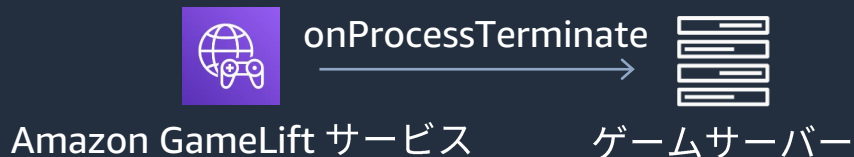


スポットインスタンスを使うために

- キューを使用してゲームセッションを配置



- ゲームサーバーで
スポットの中断を処理できるようにする



フリートメトリクス

時間経過にともなうメトリクスの変化をリアルタイムに追跡

- インスタンスの数
- ゲームセッション, プレイヤーセッション
- サーバープロセス
- インスタンスのパフォーマンス



フリート容量のスケーリング

インスタンスをスケーリングしてコスト削減とプレイヤー体験をbalancing
手動で設定するかフリートメトリクスに基づく2種類の Auto Scaling を使用



インスタンスの制限

フリート容量の許容範囲を指定します。この範囲を超えてフリート容量を手動設定、または自動スケールすることはできません。

最小インスタンス数の設定 最大インスタンス数の設定

Auto Scaling ポリシー

フリートの Auto Scaling を管理します。既定では、すべての Auto Scaling ポリシーは有効になっています。容量のバッファを維持するには、ターゲット Auto Scaling をオンにします。必要に応じて、すべてのポリシーを無効化し、目的の容量を手動で設定します。Auto Scaling の設定の詳細情報を確認してください。

手動で目的のインスタンス数を調整する ⓘ

使用できるゲームセッションの 空きゲームセッションの割合 ⓘ

フリートのすべてのスケーリングポリシーを無効にする ⓘ

スケーリングポリシー

フリートのすべての現行ポリシーを表示します。ラベルはスケーリングポリシーが無効にされた日付を示します。ターゲットベースのポリシーを管理するには、このページのバッファオプションを使用します。ルールベースのポリシーは、AWS CLI または SDK で管理する必要があります。

	名前	ポリシータイプ	ポリシー
有効	targetBasedPolicy	TargetBased	使用できるゲームセッションの 5 使用可能なゲームセッションの割合

ターゲット追跡

- フリートが維持するバッファのサイズを指定
- 多くのゲームでシンプルで効果的に機能

ルールベースのスケーリング

- フリートメトリクスを用いた評価式を作成して詳細に制御
- 特別な状況に対処するためターゲット追跡の補助として有効

Amazon GameLift の概念

Server Hosting



カスタムゲーム
サーバー



リアルタイム
サーバー

Hosting Resource



フリート



キュー



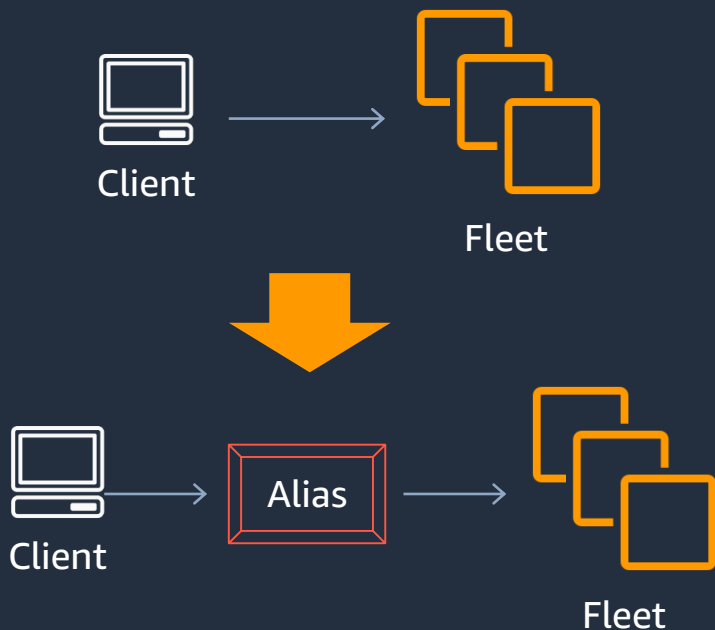
エイリアス



FlexMatch

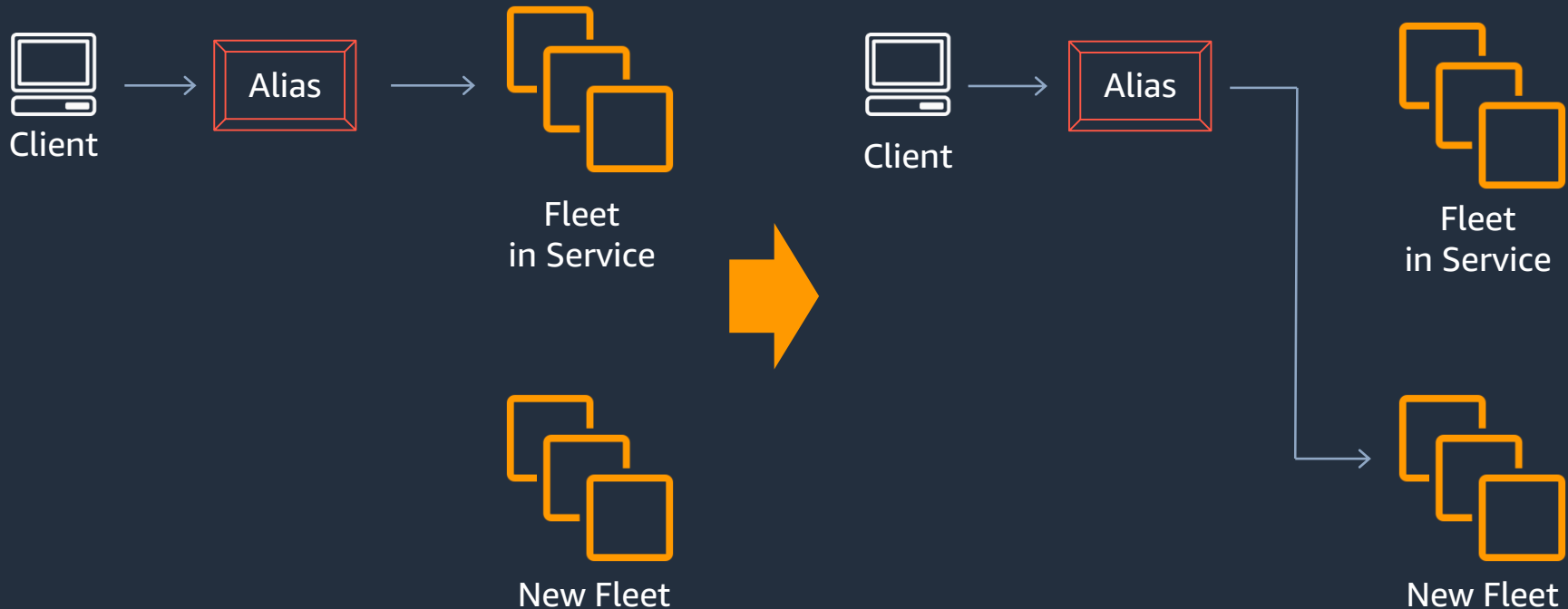
エイリアスとは

フリートもしくはエンドポイントのニックネーム
送信先を切り替える際にクライアント側の更新が不要になる

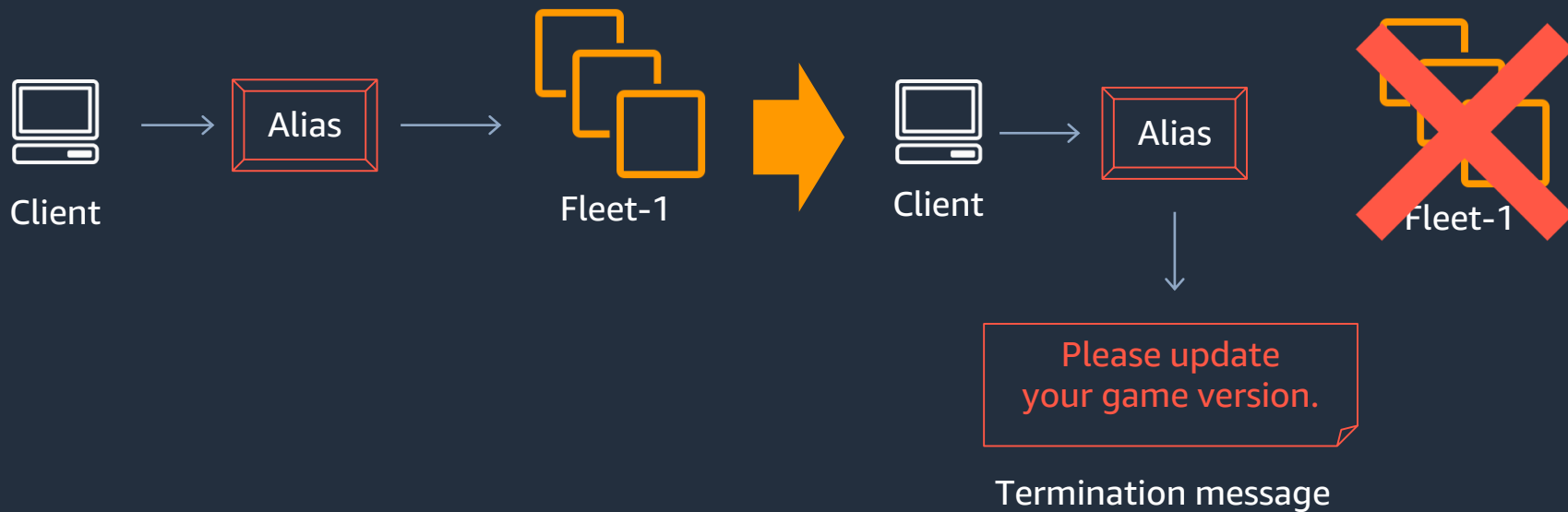


- フリート ID をエイリアス ID に置き換え
- 対象のフリートを直接指定するゲームに有効
- 2 種類のルーティング戦略
 - シンプルエイリアス
 - ターミナルエイリアス

エイリアスのルーティング戦略： シンプルエイリアス



エイリアスのルーティング戦略： ターミナルエイリアス



Amazon GameLift の概念

Server Hosting



カスタムゲーム
サーバー



リアルタイム
サーバー

Hosting Resource



フリート



エイリアス



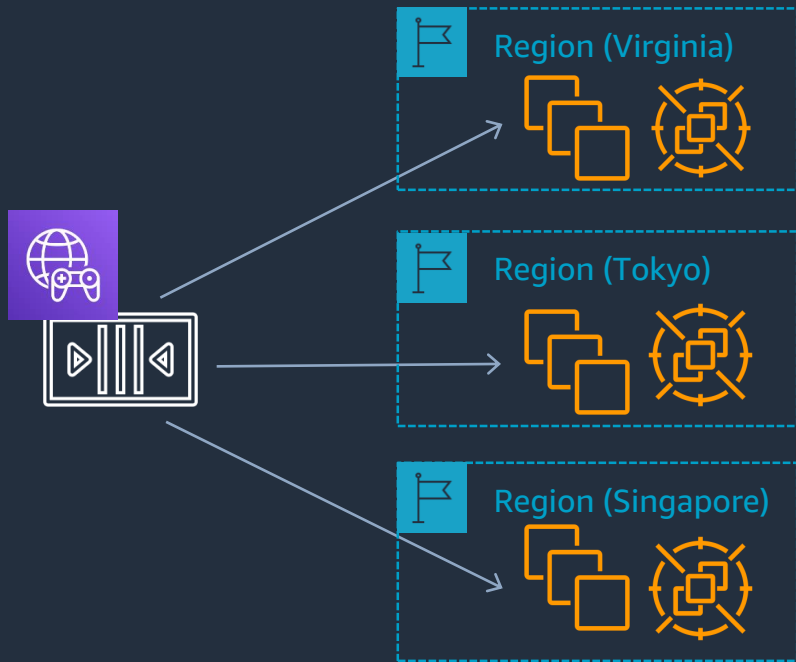
キュー



FlexMatch

Amazon GameLift におけるキューとは

ゲームセッションを配置する候補となる送信先のリスト
複数リージョンを跨いだフリートの指定でゲームセッションの配置を効率化



- プレイヤーのレイテンシーを最小限に抑える
- スポットインスタンスを活用してコスト削減
- キューのフォールバック処理とフリートの Auto Scaling 機能によりゲームセッションの配置速度と可用性が向上
- キューアクティビティのメトリクスを取得しリアルタイムにモニタリング

FleetIQ とは

可能な限り最適なフリートを検索する仕組み
レイテンシー、送信先の優先順位、コスト、スポット中断率に基づいて選択

プレイヤーレイテンシーポリシー

1. 支出 秒 すべてのプレイヤーレイテンシーが以下より小さい送信先を検索します ミリ秒、次に ... ✕
2. 支出 秒 すべてのプレイヤーレイテンシーが以下より小さい送信先を検索します ミリ秒、次に ... ✕
3. 支出 残りのキュータイムアウト すべてのプレイヤーレイテンシーが以下より小さい送信先を検索します ミリ秒。

[プレイヤーレイテンシーポリシーの追加](#)



キュー

送信先の優先順位リスト

優先度(既定)	リージョン*	タイプ*	フリート名またはエイリアス名*
1	us-east-1	フリート	fleet-a75ec7f5-8e76-4f90-bde5-4
2	us-east-1	フリート	fleet-3a6ab817-0b83-4d3c-bfe9-
3	ap-northeast-1	フリート	fleet-0c5fc707-1f36-409b-b0b9-

[送信先の追加](#)

コスト



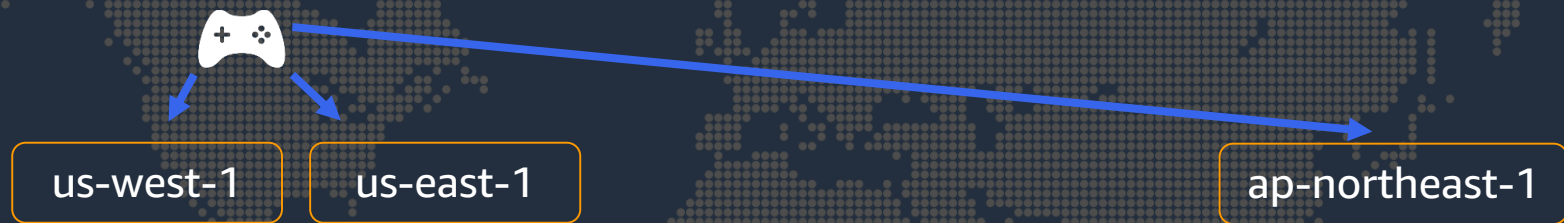
スポット中断率



最適な配置先

プレイヤーのレイテンシーデータの取得例

Amazon GameLift エンドポイントに対して Ping を送信してレイテンシーを取得

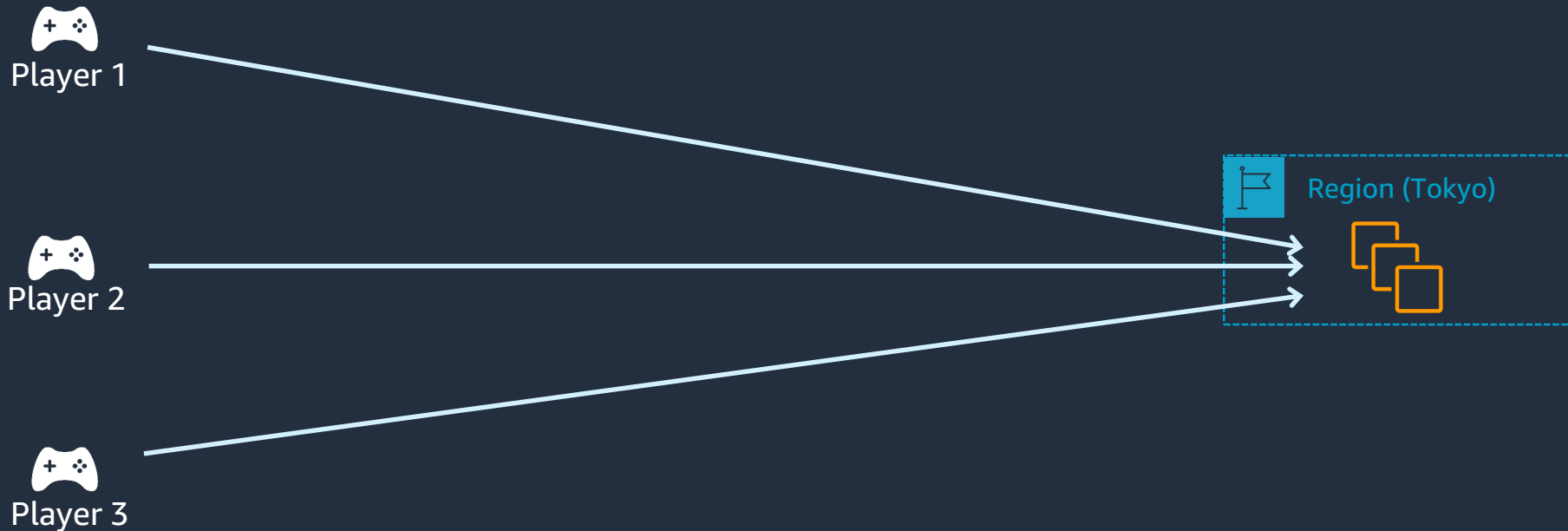


https://docs.aws.amazon.com/ja_jp/general/latest/gr/rande.html#gamelift_region

キューによるゲームセッションの配置



キューによるゲームセッションの配置



Amazon GameLift の概念

Server Hosting



カスタムゲーム
サーバー



リアルタイム
サーバー

Hosting Resource



フリート



エイリアス



キュー



FlexMatch

Amazon GameLift FlexMatch とは

カスタマイズ可能なマッチメイキングサービス
マッチングやリソースの手配を通じて最善のプレイヤー体験を実現



プレイヤーマッチングの評価方法を
ゲームに合わせてカスタマイズ

キューを使用して
最適なゲームセッションを効率的に配置

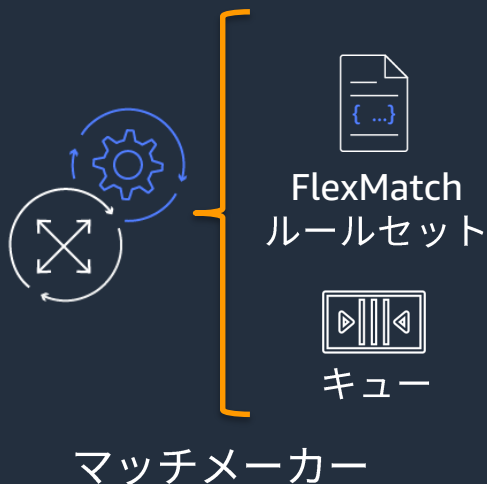
マッチメイキングのアクティビティに関する
メトリクスを収集してリアルタイムに監視

最大 200 人の大規模なマッチングが可能

マッチングされたゲームの
空きプレイヤースロットを埋める
マッチバックフィル機能を提供

(現在はカスタムゲームサーバーのみ)

FlexMatch の統合



- マッチメイキング設定を Amazon GameLift コンソールで作成
- クライアントサービスから マッチメイキングリクエストを受信して処理



- マッチメイキングリクエストを送信
- マッチメイキングのステータスを追跡
- ゲームクライアントに必要な情報を引き渡す



- Amazon GameLift サービスから渡される マッチメーカーデータを必要に応じて使用
- バックフィルリクエストを送信

マッチメイキングのプロセス (1/6)



マッチメイキングを
リクエスト

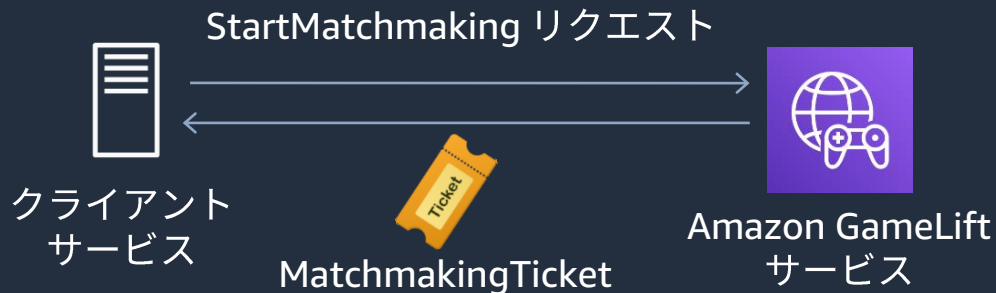


マッチング候補を検索



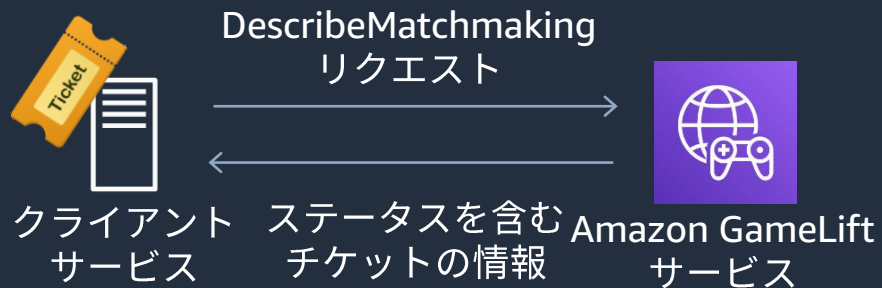
プレイヤーが承諾
(オプション)

- 使用するマッチメーカー
 - 1人以上のプレイヤー情報
 - プレイヤーのレイテンシー
- など

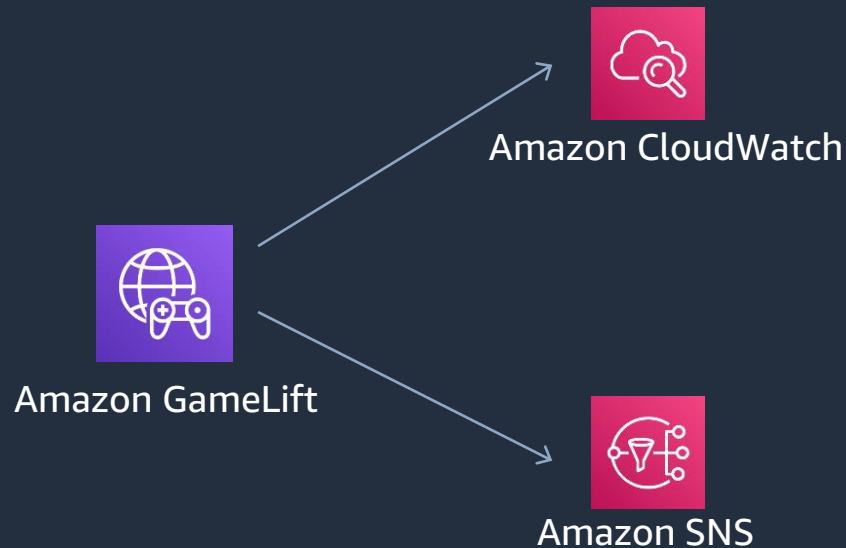


マッチメイキングリクエストのステータス追跡

継続的なポーリング



イベント通知



マッチメイキングのプロセス (2/6)



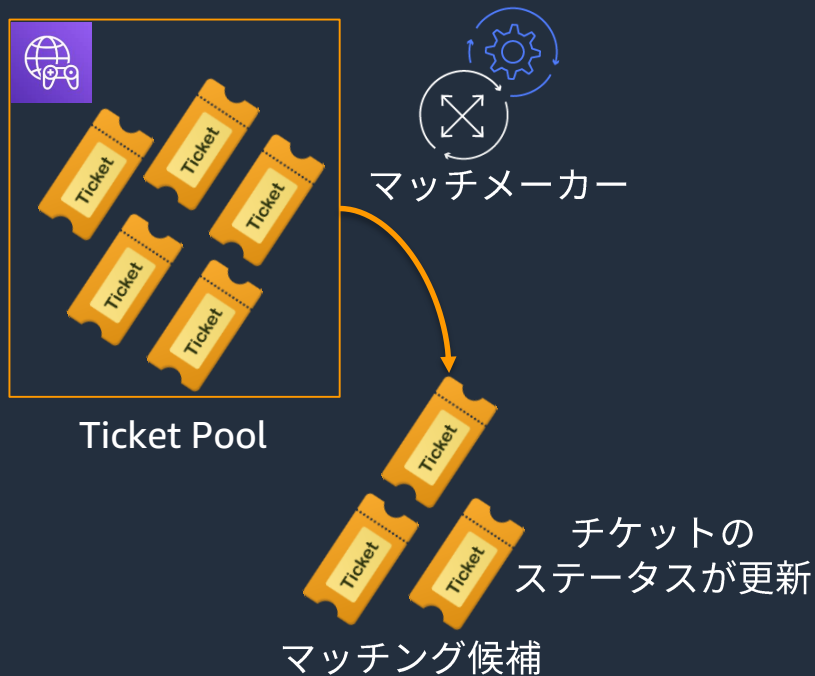
マッチメイキングを
リクエスト



マッチング候補を検索



プレイヤーが承諾
(オプション)



マッチメイキングのプロセス (3/6)



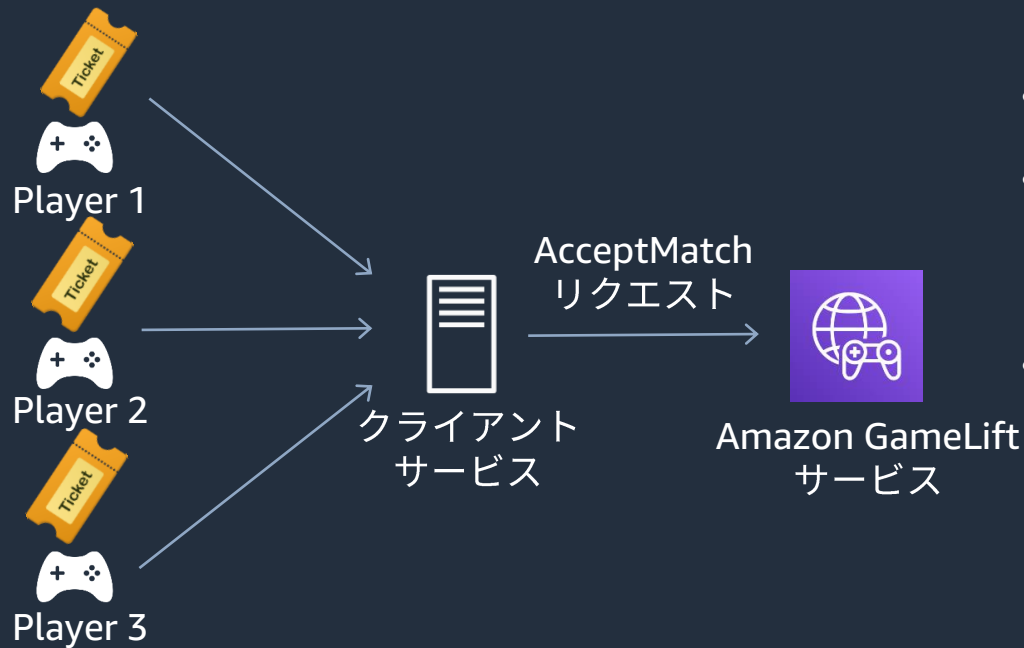
マッチメイキングを
リクエスト



マッチング候補を検索



プレイヤーが承諾
(オプション)



- 全員が承諾することでマッチング成立
- プレイヤーが1人でも却下するか制限時間内に承諾されなかった場合マッチング案は破棄される
- マッチング案が破棄された場合
 - 承諾したプレイヤー
→ Ticket Pool に戻る
 - 承諾しなかったプレイヤー
→ 処理を中断

マッチメイキングのプロセス (4/6)



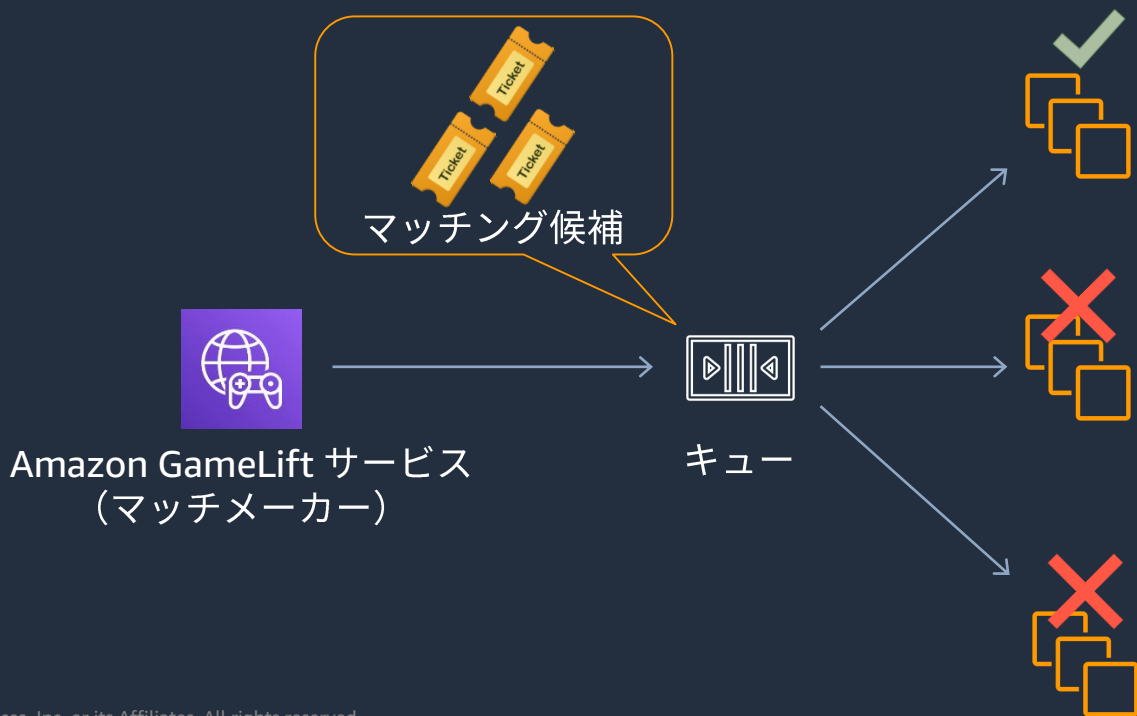
マッチングをホストする
リソースを検索



ゲームセッションを
開始



プレイヤーを
ゲームセッションに接続



マッチメイキングのプロセス (5/6)



マッチングをホストする
リソースを検索



ゲームセッションを
開始



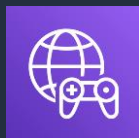
プレイヤーを
ゲームセッションに接続



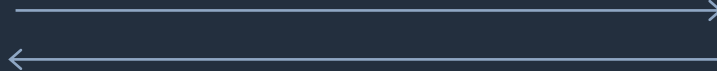
マッチメーカー
データ

使用したマッチメーカー
プレイヤーデータ
チーム情報 など

ゲームセッション開始リクエスト



Amazon GameLift
サービス



完了を通知



マッチメイキングのプロセス (6/6)



マッチングをホストする
リソースを検索



ゲームセッションを
開始



プレイヤーを
ゲームセッションに接続



Amazon GameLift
サービス

プレイヤーセッションを
作成



マッチング候補

ゲームセッションへの
接続情報が追加される

チケットの
ステータスが更新



マッチバックフィルのプロセス (1/5)



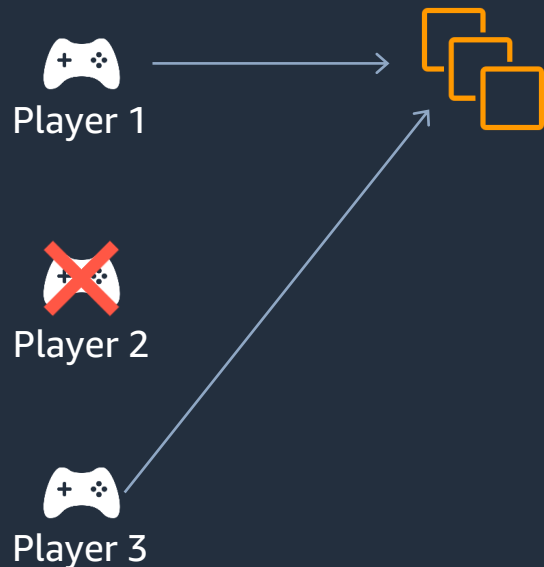
マッチバックフィルを
リクエスト



マッチング候補を検索

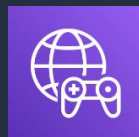


新規のプレイヤーが
承諾 (オプション)



手動

ゲームサーバーからリクエストを送信



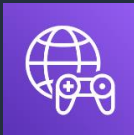
Amazon GameLift
サービス

StartMatchBackfill リクエスト

MatchmakingTicket

自動

マッチメイキング設定の自動バックフィルを有効



Amazon GameLift
サービス

空きプレイヤーロットができたなら
自動でリクエストを生成

マッチバックフィルのプロセス (2/5)



マッチバックフィルを
リクエスト



マッチング候補を検索



新規のプレイヤーが
承諾 (オプション)



Ticket Pool



マッチメーカー



マッチング候補

チケットの
ステータスが更新

- 新規のプレイヤーとバックフィルチケットを同等に評価
- マッチング候補につき1つだけバックフィルチケットが含まれる

マッチバックフィルのプロセス (3/5)



マッチバックフィルを
リクエスト



マッチング候補を検索



新規のプレイヤーが
承諾 (オプション)




マッチバックフィルのプロセス (4/5)



既存のゲームセッションを更新

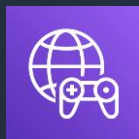


プレイヤーを
既存のゲームセッションに接続


マッチメーカー
データ

使用したマッチメーカー
プレイヤーデータ
チーム情報 など

ゲームセッション更新リクエスト



Amazon GameLift
サービス

完了を通知



Player 1



Player 3




マッチバックフィルのプロセス (5/5)



既存のゲームセッションを更新



新規のプレイヤーを
既存のゲームセッションに接続

 Amazon GameLift
サービス

プレイヤーセッションを
作成



マッチング候補

ゲームセッションへの
接続情報が追加される

チケットの
ステータスが更新



New Player



Player 1

Player 3

FlexMatch ルールセット

マッチングを構築する一連の手順 チームのサイズと構造を定義しプレイヤーの評価方法を指定

- プレイヤーのスキルレベルをマッチングに使いたい
- 2つのチームを 4~8 人で構成したい
- チームのスキルレベルの平均の差が 10 を超えないようにマッチングさせたい
- 2つのチームが同じ人数になるようにマッチングさせたい
- もしマッチングしないならスキルレベルの差の上限を緩めたい



```
{
  "name": "aliens_vs_cowboys",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "skill",
    "type": "number",
    "default": 10
  }],
  "teams": [{
    "name": "cowboys",
    "maxPlayers": 8,
    "minPlayers": 4
  }, {
    "name": "aliens",
    "maxPlayers": 8,
    "minPlayers": 4
  }],
  "rules": [{
    "name": "FairTeamSkill",
    "description": "The average skill of players in each team is within 10 points from",
    "type": "distance",
    // get skill values for players in each team and average separately to produce list
    "measurements": [ "avg(teams[*].players.attributes[skill])" ],
    // get skill values for players in each team, flatten into a single list, and average
    "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
    "maxDistance": 10 // minDistance would achieve the opposite result
  }, {
    "name": "EqualTeamSizes",
    "description": "Only launch a game when the number of players in each team matches",
    "type": "comparison",
    "measurements": [ "count(teams[cowboys].players)" ],
    "referenceValue": "count(teams[aliens].players)",
    "operation": "=" // other operations: !=, <, <=, >, >=
  }],
  "expansions": [{
    "target": "rules[FairTeamSkill].maxDistance",
    "steps": [{
      "waitTimeSeconds": 5,
      "value": 50
    }, {
      "waitTimeSeconds": 15,
      "value": 100
    }
  ]
}]
}
```

FlexMatch ルールセットのコンポーネント

コンポーネント	記述内容	定義必須
name	ルールセット構文内のラベル	
ruleLanguageVersion	プロパティ式言語のバージョン "1.0" を指定	<input type="radio"/>
playerAttributes	ルールに使用するプレイヤーの属性	
teams	チームの構造とサイズを記述。 maxPlayers 設定値 (>40) で 大規模マッチングか否か判断	<input type="radio"/>
rules	マッチングにおける プレイヤーの評価方法	<input type="radio"/>
expansions	経過時間によるマッチング条件の緩和	
algorithm	プレイヤー間の類似性重視もしくは 低レイテンシ優先のマッチングを選択	大規模なマッチングに必須

FlexMatch ルールセットの例 (1 / 2)

- プレイヤーのスキルレベルをマッチングに使いたい
→ **playerAttributes**

```
"playerAttributes": [{  
  "name": "skill",  
  "type": "number",  
  "default": 10  
}],
```

- 2つのチームを 4~8 人で構成したい
→ **teams**

```
"teams": [{  
  "name": "cowboys",  
  "maxPlayers": 8,  
  "minPlayers": 4  
}, {  
  "name": "aliens",  
  "maxPlayers": 8,  
  "minPlayers": 4  
}],
```

FlexMatch ルールセットの例 (2 / 2)

- チームのスキルレベルの平均の差が 10 を超えないようにマッチングさせたい
- 2つのチームが同じ人数になるようにマッチングさせたい
- もしマッチングしないならスキルレベルの差の上限を緩めたい
→ **expansions**

→ **rules**

```
"rules": [{
  "name": "FairTeamSkill",
  "description": "The average skill of players in each team is within 10",
  "type": "distance",
  // get skill values for players in each team and average separately to
  "measurements": [ "avg(teams[*].players.attributes[skill])" ],
  // get skill values for players in each team, flatten into a single list
  "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
  "maxDistance": 10 // minDistance would achieve the opposite result
}, {
  "name": "EqualTeamSizes",
  "description": "Only launch a game when the number of players in each team is equal",
  "type": "comparison",
  "measurements": [ "count(teams[cowboys].players)" ],
  "referenceValue": "count(teams[aliens].players)",
  "operation": "=" // other operations: !=, <, <=, >, >=
}],
```

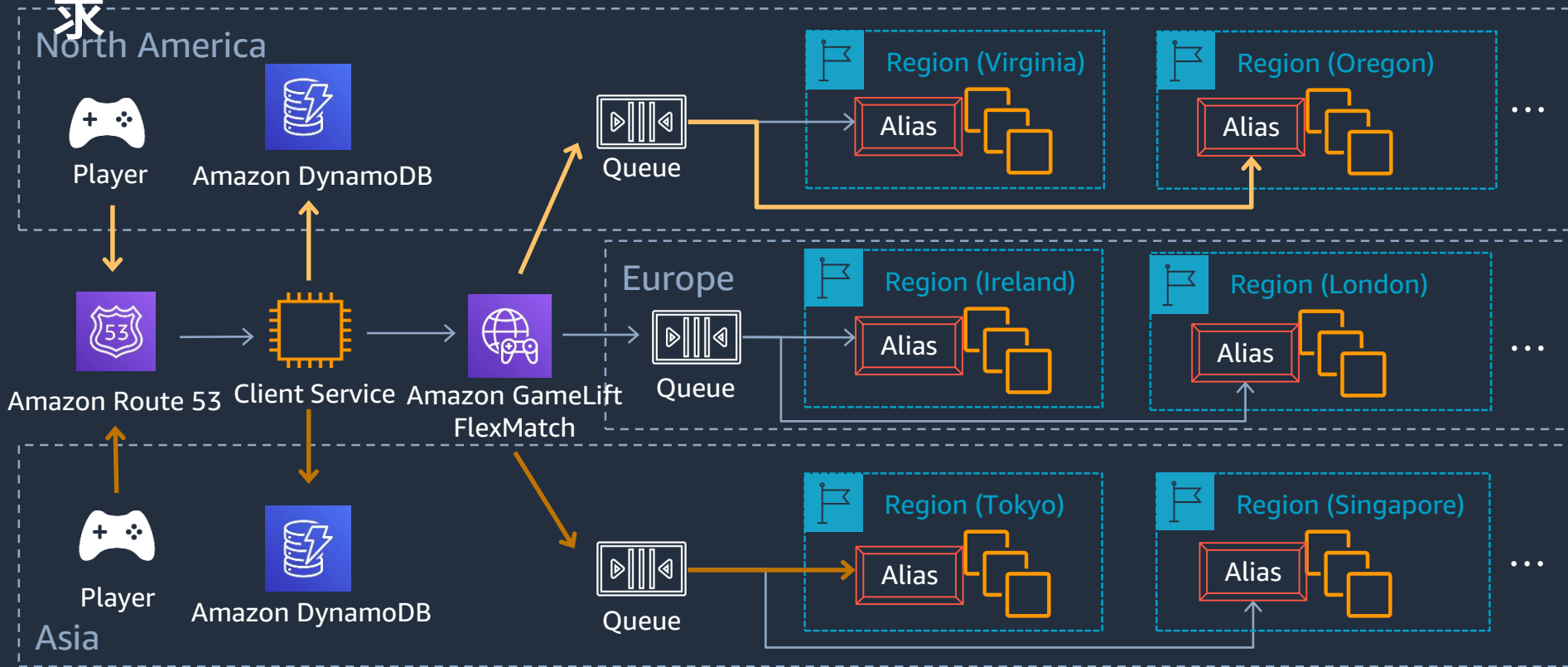
```
"expansions": [{
  "target": "rules[FairTeamSkill].maxDistance",
  "steps": [{
    "waitTimeSeconds": 5,
    "value": 50
  }, {
    "waitTimeSeconds": 15,
    "value": 100
  }]
}]
```

本日のアジェンダ

- マルチプレイヤーゲームについて
- Amazon GameLift とは
- Amazon GameLift によるサーバーのホスティング
- Amazon GameLift ホスティングリソース
- Amazon GameLift の応用例
- 補足情報やご利用料金について
- まとめ

GameLift グローバルルーティング – 低レイテンシーの追

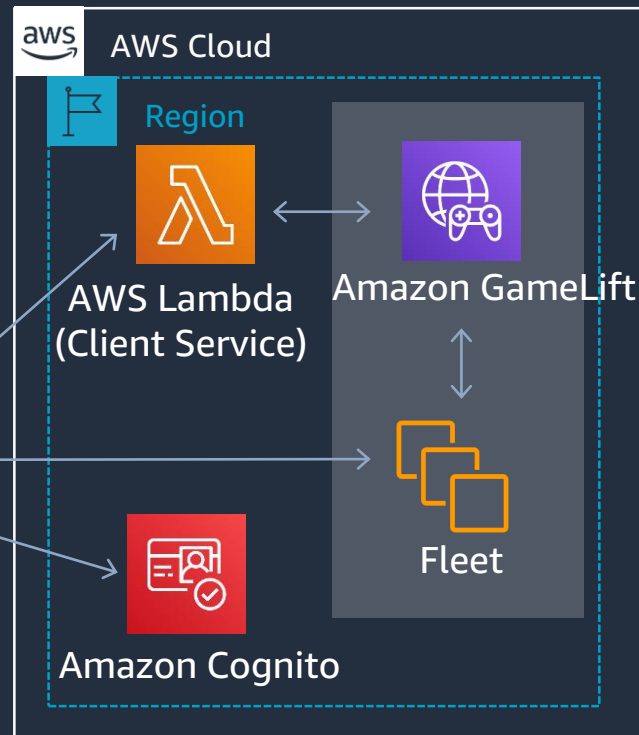
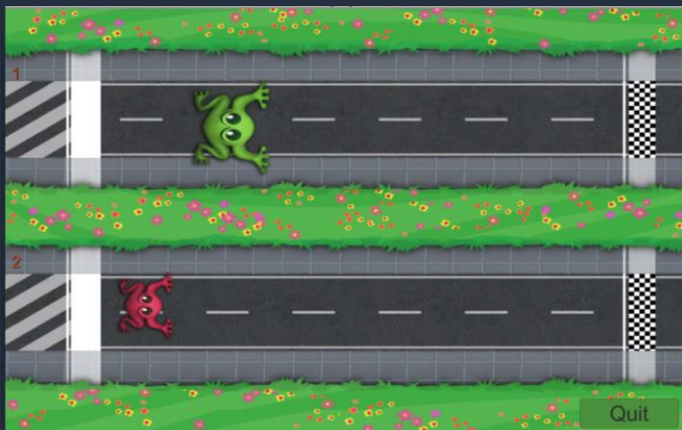
求



本日のアジェンダ

- マルチプレイヤーゲームについて
- Amazon GameLift とは
- Amazon GameLift によるサーバーのホスティング
- Amazon GameLift ホスティングリソース
- Amazon GameLift の応用例
- 補足情報やご利用料金について
- まとめ

Amazon GameLift リアルタイムサーバーを試す



<https://aws.amazon.com/jp/blogs/news/creating-servers-for-multiplayer-mobile-games-with-amazon-gamelift/>

ご利用料金について

- 従量課金制
- 稼働しているインスタンス（ゲームサーバー）の使用時間
 - Auto Scaling とスポットインスタンスでコストを削減
- インスタンスからインターネットへのデータ転送量
 - ゲームクライアントとゲームサーバー間のアウトバウンドデータ転送に対して課金



- AWS 無料利用枠
 - Amazon GameLift 向け
c4.large オンデマンドインスタンスの使用 125 時間/月
EBS 汎用 (SSD) ストレージ 50 GB

本日のアジェンダ

- マルチプレイヤーゲームについて
- Amazon GameLift とは
- Amazon GameLift によるサーバーのホスティング
- Amazon GameLift ホスティングリソース
- Amazon GameLift の応用例
- 補足情報やご利用料金について
- **まとめ**

まとめ



- AWS クラウドで専用ゲームサーバーをグローバルにホスティング
- 柔軟なマッチメイキングで最善のプレイヤー体験を実現
- マルチプレイヤーゲーム開発の多くの技術的なりスクを低減



ゲーム開発を加速

Q&A

お答えできなかったご質問については

AWS Japan Blog 「<https://aws.amazon.com/jp/blogs/news/>」にて
後日掲載します。

AWS の日本語資料の場所「AWS 資料」で検索



日本担当チームへお問い合わせ サポート 日本語 ▼ アカウント ▼

コンソールにサインイン

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 🔍

AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

[AWS Webinar お申込 »](#)

[AWS 初心者向け »](#)

[業種・ソリューション別資料 »](#)

[サービス別資料 »](#)

<https://amzn.to/JPArchive>



AWS Well-Architected 個別技術相談会

毎週”W-A個別技術相談会”を実施中

- AWSのソリューションアーキテクト(SA)に
対策などを相談することも可能

- **申込みはイベント告知サイトから**
(<https://aws.amazon.com/jp/about-aws/events/>)

AWS イベント で[検索]



ご視聴ありがとうございました

AWS 公式 Webinar
<https://amzn.to/JPWebinar>



過去資料
<https://amzn.to/JPArchive>

