aws

Guida all'hosting

Amazon GameLift Servers

Amazon GameLift Servers: Guida all'hosting

Table of Contents

Cos'è Amazon GameLift Servers?

Amazon GameLift ServersUtilizzalo per distribuire, gestire e scalare server dedicati a basso costo nel cloud per giochi multiplayer basati su sessioni. Basato su un'infrastruttura informatica AWS globale, Amazon GameLift Servers aiuta a fornire server di gioco ad alte prestazioni e alta affidabilità, scalando dinamicamente l'utilizzo delle risorse per soddisfare la domanda dei giocatori in tutto il mondo.

Cosa puoi fare con Amazon GameLift Servers

Amazon GameLift Serverssupporta questi casi d'uso e molto altro ancora:

- Ospita i tuoi server di gioco multiplayer personalizzati nel cloud con EC2 hosting Amazon GameLift Servers gestito.
- Esegui risorse di hosting gestito a basso costo utilizzando le istanze Spot di Amazon Elastic Compute Cloud EC2 (Amazon).
- Ospita i tuoi server di gioco containerizzati per avere flessibilità su tutte le piattaforme e supportare le migrazioni con container gestiti. Amazon GameLift Servers
- Crea una soluzione di hosting ibrida per supportare l'hosting and/or on-premise multi-cloud gestendo al contempo le sessioni di gioco in un unico posto con Anywhere. Amazon GameLift Servers
- Crea un solido sistema di matchmaking per i tuoi giochi multiplayer con. Amazon GameLift Servers FlexMatch
- Scala automaticamente la tua capacità di hosting gestito per soddisfare le tue esigenze di gioco in base all'utilizzo effettivo dei giocatori.
- Gestisci le tue risorse di EC2 calcolo Amazon per i giochi in un unico posto utilizzando Amazon GameLift ServersFleetIQ.
- Crea un ambiente di test iterativo per le build di server e client di gioco con Amazon GameLift Servers Anywhere and. EC2
- Per i giochi che non richiedono un server di gioco personalizzato, configura una soluzione server leggera con. Amazon GameLift Servers Realtime

> **Tip**
>
> Inizia subito con Amazon GameLift Servers

Come utilizzare Amazon GameLift Servers

Utilizzare questi strumenti per l'utilizzo di Amazon GameLift Servers.

AWS CLI

Usa il AWS Command Line Interface (AWS CLI) per effettuare chiamate all' AWS SDK, inclusa l'API del servizio perAmazon GameLift Servers. Vedi Guida introduttiva alla AWS CLI Guida per l'AWS Command Line Interface utente.

Console Amazon GameLift Servers

Utilizza il form AWS Management ConsoleAmazon GameLift Serversper configurare le risorse, gestire le distribuzioni dei server di gioco e tenere traccia delle metriche di prestazioni e utilizzo. La Amazon GameLift Servers console è un'alternativa alla GUI alla gestione delle risorse a livello di codice o con. AWS CLI

Amazon GameLift Servers SDKs

Amazon GameLift Servers SDKs Contengono le librerie necessarie per stabilire la comunicazione tra i client di gioco, i server di gioco e i servizi di gioco e il servizio. Amazon GameLift Servers Per ulteriori informazioni, consulta Ottieni strumenti Amazon GameLift Servers di sviluppo.

Client SDK per Amazon GameLift Servers Realtime

Il client SDK for ti Amazon GameLift Servers Realtime consente di connettere il tuo client di gioco a un Realtime server fornito daAmazon GameLift Servers, partecipare a sessioni di gioco e rimanere sincronizzato con altri giocatori. Scarica l'SDK e scopri di più su come effettuare chiamate API con l'API Amazon GameLift ServersRealtime client (C#).

Amazon GameLift Serverssoluzioni

Amazon GameLift Serversoffre una gamma di soluzioni per gli sviluppatori che stanno creando giochi multiplayer basati su sessioni.

Soluzioni per sviluppatori di giochi

- Amazon GameLift Serversopzioni di hosting
- Amazon GameLift ServersFlexMatchper il matchmaking
- Amazon GameLift ServersFleetIQper l'hosting Amazon EC2 autogestito
- Amazon GameLift ServersRealtimecon logica server personalizzabile

Amazon GameLift Serversopzioni di hosting

Quando lavori con Amazon GameLift Servers i tuoi server di gioco, hai diverse opzioni su dove e come sono ospitati i tuoi server di gioco. Sia che tu voglia utilizzare le risorse di hosting che già possiedi o configurare un hosting basato su cloud gestito daAmazon GameLift Servers, puoi creare un'esperienza di hosting senza interruzioni per i tuoi giocatori.

Gestito EC2

Contenitori gestiti

Hosting ibrido

Hosting ovunque

Gestito EC2

Con l' EC2 hosting Amazon GameLift Servers gestito, puoi scaricare la maggior parte del lavoro di gestione dei tuoi server di gioco. Scegli le risorse di elaborazione tra un'ampia selezione di tipi di EC2 istanze Amazon. Integra i tuoi progetti di gioco e lascia che siano i dettagli a Amazon GameLift Servers occuparsene. Per ulteriori informazioni sull'hosting gestito, consultaFunzionamento di Amazon GameLift Servers.

Inizia a sviluppare una soluzione di hosting Amazon GameLift Servers gestito per il tuo gioco.

Funzionalità principali

- Ospita giochi multiplayer eseguiti su sistemi operativi Amazon Linux o Windows Server.
- Offri esperienze di gioco a bassa latenza ai tuoi giocatori, ovunque si trovino. Distribuisci server di gioco a livello globale in tutte Regioni AWS le Local Zones Amazon GameLift Servers supportate. Per un elenco completo, consulta Amazon GameLift Serverssedi di assistenza.
- Usa il posizionamento Amazon GameLift Servers intelligente delle sessioni di gioco in modo che i giocatori ottengano sempre la migliore esperienza possibile come giocatore ospitato. Puoi fare

affidamento sul Amazon GameLift Servers processo decisionale oppure puoi personalizzare in base a criteri di posizionamento come costo, latenza dei giocatori e posizioni geografiche.

- Scegli come scalare le tue risorse di hosting per soddisfare la domanda dei giocatori. Gestisci la capacità manualmente o imposta la scalabilità automatica. Con la scalabilità automatica basata sull'obiettivo, puoi mantenere un buffer di capacità inattiva di dimensioni dinamiche, che ti aiuta a controllare i costi garantendo al contempo che i nuovi giocatori possano entrare in gioco con tempi di attesa minimi.
- Consenti di Amazon GameLift Servers distribuire e gestire i tuoi server di gioco basati sul cloud. Amazon GameLift Serverscrea le risorse necessarie, installa il software del server di gioco e avvia automaticamente i processi per ospitare sessioni di gioco per i giocatori. Imposta un monitoraggio sanitario personalizzato e consenti di Amazon GameLift Servers rilevare e risolvere le risorse con scarse prestazioni.
- Sfrutta le funzionalità di Amazon GameLift Servers monitoraggio per valutare le prestazioni e l'utilizzo. Puoi tenere traccia delle metriche su fattori quali le prestazioni dell'hardware, l'efficienza del posizionamento delle sessioni di gioco e i cicli di vita dei processi del server. Puoi tenere traccia delle sessioni di gioco attive e delle sessioni dei giocatori per osservarne l'utilizzo nel tempo. Puoi anche scaricare e archiviare i registri delle sessioni di gioco.
- Per l'hosting di produzione, automatizza la gestione e la distribuzione delle risorse di hosting di giochi utilizzando AWS CloudFormation modelli per e the. Amazon GameLift Servers AWS Cloud Development Kit (AWS CDK) Sfrutta strumenti e servizi di integrazione continua e distribuzione continua (CI/CD) come. AWS CodePipeline

Contenitori gestiti

Amazon GameLift Serversfornisce una soluzione completa di cloud hosting per server di gioco containerizzati. Con i container Amazon GameLift Servers gestiti, puoi sfruttare i vantaggi principali dell'utilizzo dei container, come portabilità, agilità e tolleranza agli errori. Le seguenti funzionalità sono disponibili con le flotte di container gestite.

Inizia a sviluppare una soluzione di hosting Amazon GameLift Servers gestito per il tuo server di gioco containerizzato.

Funzionalità principali

- Sviluppa un'architettura personalizzata con contenitori leggeri per eseguire il software del tuo server di gioco su risorse di hosting basate su Amazon GameLift Servers Linux.

- Usa gli strumenti Docker per creare un'immagine di contenitore basata su Linux. Archivia le immagini per la distribuzione in un repository Amazon Elastic Container Registry (Amazon ECR).
- Offri ai giocatori esperienze a bassa latenza distribuendo le risorse della flotta di container in qualsiasi Regione AWS zona locale che supporti. Amazon GameLift Servers Consultare Amazon GameLift Serverssedi di assistenza.
- Gestisci il ciclo di vita della flotta con strumenti per modellare le versioni dei server di gioco e distribuire gli aggiornamenti della flotta.
- Usa le funzionalità di posizionamento delle sessioni di Amazon GameLift Servers gioco, tra cui code e FlexMatch matchmaking, per trovare le migliori partite di sessione di gioco possibili per i tuoi giocatori.
- Metti alla prova l'architettura del server di gioco e del container con il Amazon GameLift Servers servizio utilizzando una flotta Anywhere. Testa il gioco localmente o in un ambiente di test basato sul cloud.
- Tieni traccia delle prestazioni di hosting dei giochi con metriche prestazionali specifiche del contenitore. Monitora lo stato delle risorse della tua flotta utilizzando metriche hardware.
- Gestisci le risorse della flotta di container utilizzando AWS CloudFormation modelli perAmazon GameLift Servers.

Hosting ibrido

Utilizza il Amazon GameLift Servers servizio con una combinazione di hosting Amazon GameLift Servers gestito e hosting autogestito Anywhere. Un approccio ibrido ti consente di creare la soluzione di cui hai bisogno in questo momento e allo stesso tempo di prepararti per dove dovrai trovarti in futuro. Gli scenari comuni in cui una soluzione ibrida ha senso includono:

- Espandi la tua soluzione di hosting a Cloud AWS. Completa le funzionalità della tua soluzione di hosting esistente (hardware locale o altro hosting basato sul cloud) aggiungendo l'Amazon GameLift Servershosting gestito. Con l'hosting gestito, puoi aumentare la capacità di hosting o aggiungere capacità «burst» per scalare rapidamente e pagare solo le risorse quando ne hai bisogno. Puoi anche sfruttare l'impronta globale del Amazon GameLift Servers servizio per raggiungere più giocatori in tutto il mondo e offrire l'esperienza multiplayer a bassa latenza che si aspettano.
- Preparati per la migrazione all'hosting basato sul cloud. Se state considerando o avete intenzione di migrare a Cloud AWS (anziché aggiornare il vostro hardware), una soluzione di hosting ibrido è un modo valido per effettuare la transizione con la gradualità necessaria.

- Aumenta la latenza per i giocatori in località diverse da quelle servite da. Amazon GameLift Servers Se utilizzi già l'hosting Amazon GameLift Servers gestito, potresti dover supportare i giocatori in determinate situazioni. Ad esempio, potresti voler raggiungere giocatori in località insolitamente remote o ridurre significativamente la latenza in quelle aree. Aggiungi sedi di hosting personalizzate e utilizza Amazon GameLift Servers Anywhere per gestirle insieme alle risorse di hosting gestite.

Inizia a sviluppare una soluzione di hosting Amazon GameLift Servers ibrida per il tuo gioco.

Funzionalità principali

- Utilizza gli stessi componenti client e server di gioco con risorse di hosting gestite e autogestite. Offri ai giocatori un'esperienza unificata su tutte le risorse di hosting.
- Usa gli stessi FlexMatch matchmaker per piazzare le partite su tutte le risorse di hosting.
- Gestisci centralmente le tue risorse di hosting ibrido insieme mentre le distribuisci in tutto il mondo.
- Man mano che la domanda dei giocatori oscilla, gestisci senza problemi i carichi delle sessioni di gioco tra risorse gestite e autogestite.
- Con l'Amazon GameLift ServersAgent, puoi utilizzare gli stessi strumenti per gestire i cicli di vita dei server di gioco su tutti i tipi di risorse di hosting.
- Raccogli le metriche e i registri di gioco e giocatori su tutte le risorse di hosting. Sfrutta Amazon GameLift Servers le funzionalità e gli altri AWS servizi per combinare i dati e sviluppare soluzioni di osservabilità coerenti.

Hosting ovunque

Usa le flotte Amazon GameLift Servers Anywhere con gestione delle sessioni di Amazon GameLift Servers gioco, incluso il matchmaking, per ospitare i tuoi server di gioco personalizzati ovunque tu voglia. Anywherele flotte sono particolarmente utili come ambienti di test per lo sviluppo rapido e iterativo di giochi. Configura una Anywhere flotta per la tua workstation locale o un set di risorse di hosting basate su cloud. Per l'hosting di produzione, potresti utilizzare un approccio ibrido con Anywhere flotte per l'hardware locale integrate da flotte gestite. Amazon GameLift Servers

Per ulteriori informazioni sui test con Anywhere, consulta. Configura i test locali con Amazon GameLift Servers Anywhere Per ulteriori informazioni sulla configurazione di una flotta Anywhere, consultaConfigurazione di una flotta di hosting con Amazon GameLift Servers.

Inizia a sviluppare una soluzione di hosting Amazon GameLift Servers Anywhere per il tuo gioco.

Funzionalità principali

- Esegui test rapidi e iterativi mentre sviluppi i tuoi giochi multiplayer.
- Usa Amazon GameLift Servers gli strumenti per gestire i server di gioco ospitati sul tuo hardware.
- Sfrutta l'hardware disponibile più vicino ai tuoi giocatori, ovunque si trovino.

Amazon GameLift ServersFlexMatchper il matchmaking

FlexMatchUsalo per creare set di regole personalizzati per definire le partite multiplayer per il tuo gioco. FlexMatchutilizza set di regole per confrontare i giocatori compatibili per ogni partita e offrire ai giocatori l'esperienza multiplayer ideale.

Per ulteriori informazioni suFlexMatch, vedi Cos'è Amazon GameLift ServersFlexMatch?

Funzionalità principali

- Bilancia la velocità di creazione delle partite e la qualità della partita.
- Abbina giocatori o squadre in base a caratteristiche definite.
- Definisci le regole per piazzare i giocatori nelle partite in base alla latenza.

Amazon GameLift ServersFleetIQper l'hosting Amazon EC2 autogestito

Usa i gruppi di server di FleetIQ gioco per lavorare direttamente con le tue risorse di hosting in Amazon EC2 e Amazon EC2 Auto Scaling. Ciò offre il vantaggio delle Amazon GameLift Servers ottimizzazioni per un hosting di giochi economico e resiliente. Questa soluzione è destinata agli sviluppatori di giochi che necessitano di maggiore flessibilità rispetto a quella offerta dalle Amazon GameLift Servers soluzioni completamente gestite.

Per informazioni su come FleetIQ funziona con Amazon EC2 e EC2 Auto Scaling per l'hosting di giochi, consulta la Guida per gli Amazon GameLift ServersFleetIQsviluppatori.

Funzionalità principali

- Ottieni un bilanciamento ottimizzato delle istanze Spot utilizzando l'FleetIQalgoritmo.
- Utilizza le funzionalità di routing dei giocatori per gestire in modo efficiente le risorse del server di gioco e offrire ai giocatori un'esperienza migliore per partecipare alle partite.
- Ridimensiona automaticamente la capacità di hosting in base all'utilizzo dei giocatori.

- Gestisci direttamente EC2 le istanze Amazon da solo. Account AWS
- Usa uno qualsiasi dei formati eseguibili dei server di gioco supportati, inclusi Windows, Linux, container e Kubernetes.

Amazon GameLift ServersRealtimecon logica server personalizzabile

Utilizza Realtime i server per ospitare giochi che non richiedono un server di gioco personalizzato. Questa soluzione server leggera offre server di gioco che puoi configurare per adattarli al tuo gioco. Puoi ospitare Realtime server utilizzando una soluzione di hosting Amazon GameLift Servers gestito.

Per ulteriori informazioni sull'hosting con Amazon GameLift ServersRealtime, consultaIntegrazione di giochi con Amazon GameLift ServersRealtime.

Funzionalità principali

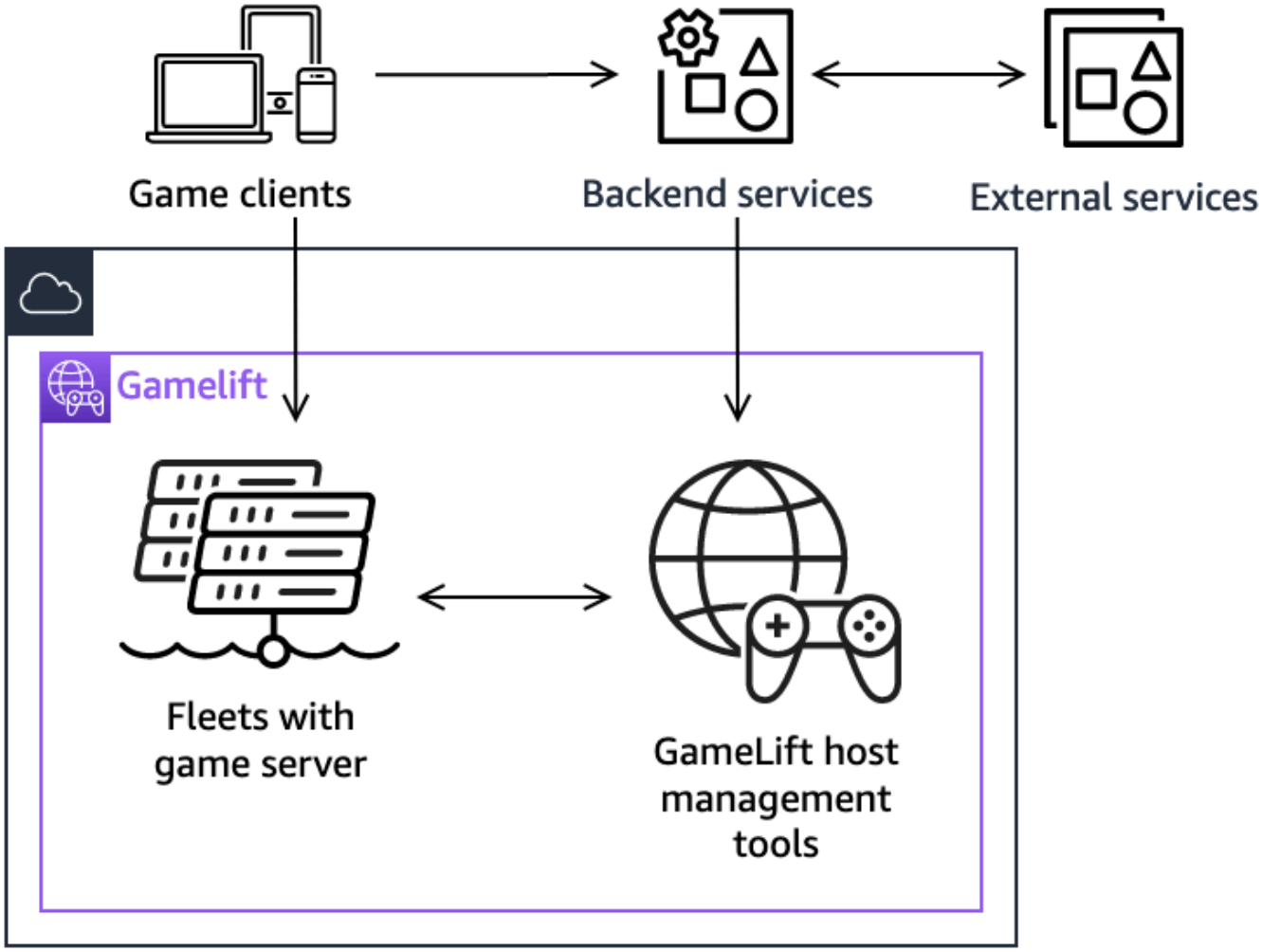
- Usa le funzionalità di Amazon GameLift Servers gestione, tra cui il ridimensionamento automatico, le code in più sedi e il posizionamento delle sessioni di gioco.
- Utilizza le risorse Amazon GameLift Servers di hosting e scegli il tipo di hardware AWS informatico per le tue flotte.
- Sfrutta uno stack di rete completo per l'interazione tra client di gioco e server.
- Ottenere funzionalità del server di gioco integrata con logica server personalizzabile.
- Eseguire aggiornamenti in tempo reale alle configurazioni Realtime e alla logica del server.

Architettura della Amazon GameLift Servers soluzione gestita

I diagrammi di questo argomento descrivono come è strutturata una soluzione di hosting completa conAmazon GameLift Servers.

Componenti di gioco con hosting

Il diagramma seguente illustra come i componenti chiave di una soluzione di Amazon GameLift Servers hosting gestito interagiscono per eseguire server di gioco dedicati e aiutare i giocatori a trovare e connettersi alle sessioni di gioco ospitate. La soluzione di hosting che svilupperai per il tuo gioco includerà la maggior parte o tutti questi componenti.



I componenti chiave di questa architettura includono quanto segue:

Client di gioco

Un client di gioco è un software in esecuzione sul dispositivo di un giocatore. Il giocatore gioca al tuo gioco partecipando a una sessione di gioco su un server di gioco ospitato. Un client di gioco chiede di partecipare a una sessione di gioco tramite un servizio di backend, riceve informazioni di connessione per una sessione di gioco e le utilizza per connettersi direttamente alla sessione di gioco. Per ulteriori informazioni, consulta Preparazione dei giochi per Amazon GameLift Servers. Quando si connette a un Realtime server, un client di gioco A utilizza l'SDK del client per. Amazon GameLift Servers Realtime

Servizi di backend

Un servizio di backend è un servizio personalizzato creato per gestire la comunicazione con il Amazon GameLift Servers servizio per conto di un client di gioco. Puoi anche utilizzare i servizi di backend per attività specifiche del gioco come l'autenticazione e l'autorizzazione dei giocatori, l'inventario o il controllo della valuta. Un servizio di backend comunica con il Amazon GameLift Servers servizio utilizzando le operazioni API nell'SDK. AWS

Un servizio di backend effettua richieste per ottenere informazioni sulla sessione di gioco esistente e per avviare sessioni di gioco. Le richieste di nuove sessioni di gioco definiscono determinate caratteristiche, come il numero massimo di giocatori. Queste richieste richiedono l'avvio del processo Amazon GameLift Servers di posizionamento della sessione di gioco. Quando una sessione di gioco è pronta per accettare giocatori, il servizio di backend recupera le informazioni di connessione e le fornisce al client di gioco.

Servizi esterni

Il gioco può contare su servizi esterni, ad esempio per la convalida di un abbonamento. Un servizio esterno può trasmettere informazioni ai tuoi server di gioco tramite un servizio di backend e. Amazon GameLift Servers

Server di gioco

Un server di gioco è il software del server di gioco che funziona su un set di risorse di hosting. Carichi il software del server di gioco suAmazon GameLift Servers, che lo distribuisce sulle risorse di hosting e avvia l'esecuzione dei processi del server. Ogni processo del server di gioco si connette al Amazon GameLift Servers servizio per segnalare la disponibilità a ospitare sessioni di gioco. Interagisce con il servizio per avviare sessioni di gioco, convalidare i nuovi giocatori connessi e segnalare lo stato delle sessioni di gioco e delle connessioni dei giocatori.

I server di gioco personalizzati comunicano con loro Amazon GameLift Servers utilizzando il server SDK for. Amazon GameLift Servers Per ulteriori informazioni, consultaIntegra giochi con server di gioco personalizzati. Realtimei server sono server di gioco forniti daAmazon GameLift Servers. Puoi personalizzare la logica del server fornendo uno script personalizzato. Per ulteriori informazioni, consulta Integrazione di giochi con Amazon GameLift ServersRealtime.

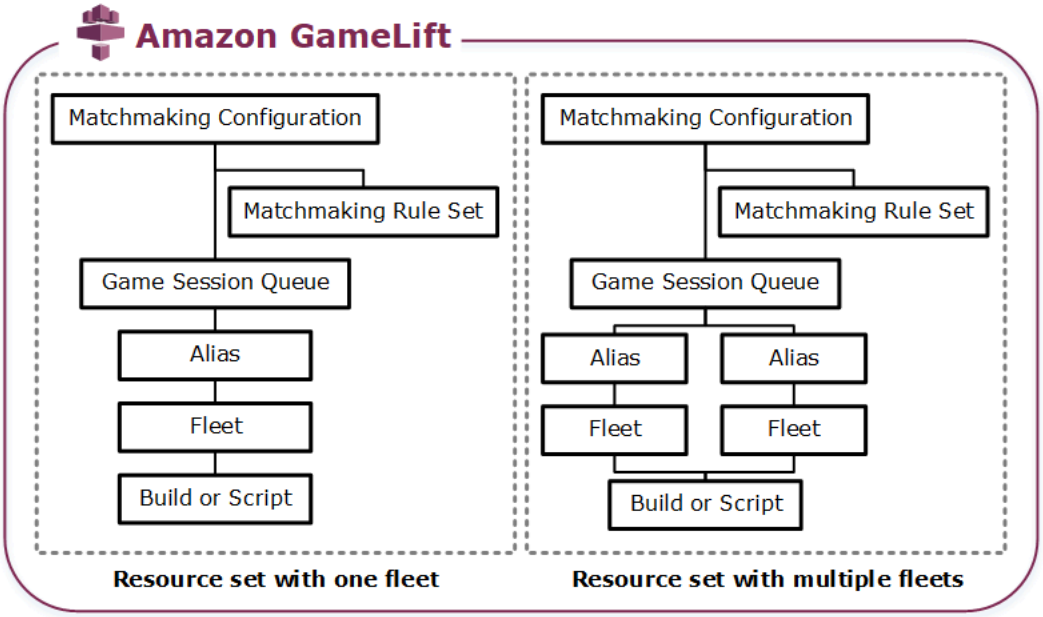
Strumenti di gestione degli host

Durante la configurazione e la gestione delle risorse di hosting, i proprietari dei giochi utilizzano strumenti di gestione dell'hosting per gestire le build o gli script dei server di gioco, le flotte, il matchmaking e le code. Il set di Amazon GameLift Servers strumenti nell' AWS SDK e nella

console offre diversi modi per gestire le risorse di hosting. Puoi accedere in remoto a qualsiasi server di gioco individuale per la risoluzione dei problemi.

Risorse per soluzioni di hosting

Il diagramma seguente illustra Amazon GameLift Servers le risorse che costituiscono una soluzione di hosting gestito. Fornisci una build di server o uno Amazon GameLift Servers Realtime script personalizzati, distribuisci una flotta di computer per ospitare i server di gioco, quindi configura una coda per le sessioni di gioco per trovare le risorse di hosting disponibili e iniziare nuove sessioni di gioco. Per i giochi che utilizzano il FlexMatch matchmaking, aggiungi una configurazione di matchmaking e un set di regole di matchmaking per generare partite tra giocatori.



Codice del server di gioco

- Build: il tuo software per server di gioco personalizzato che funziona Amazon GameLift Servers e ospita sessioni di gioco per i tuoi giocatori. Una build di gioco rappresenta l'insieme di file che eseguono il server di gioco su un particolare sistema operativo e con cui devi integrarti. Amazon GameLift Servers Carica i file di build del gioco Amazon GameLift Servers nel Regioni AWS punto in cui intendi creare le flotte. Per ulteriori informazioni, consulta Implementa una build di server personalizzata per Amazon GameLift Servers hosting.
- Script: la tua configurazione e la logica di gioco personalizzata da utilizzare con Amazon GameLift ServersRealtime. Configura Amazon GameLift Servers Realtime per i tuoi client di gioco creando uno script utilizzando JavaScript e aggiungi una logica di gioco personalizzata

per ospitare sessioni di gioco per i tuoi giocatori. Per ulteriori informazioni, consulta Implementa uno script per Amazon GameLift ServersRealtime.

Parco istanze

Una raccolta di risorse di calcolo che gestiscono i tuoi server di gioco e ospitano sessioni di gioco per i tuoi giocatori. Per informazioni su dove puoi schierare le flotte, consulta. Amazon GameLift Serverssedi di assistenza Per informazioni sulla creazione di flotte, consulta. Configurazione di una flotta di hosting con Amazon GameLift Servers

Alias

Un identificatore astratto per una flotta che puoi usare per modificare la flotta a cui i giocatori sono collegati in qualsiasi momento. Per ulteriori informazioni, consulta Crea un Amazon GameLift Servers alias.

Coda delle sessioni di gioco

Un meccanismo di posizionamento delle sessioni di gioco che riceve le richieste di nuove sessioni di gioco e cerca i server di gioco disponibili per ospitare le nuove sessioni. Per ulteriori informazioni sulle code delle sessioni di gioco, consulta. Gestione del posizionamento delle sessioni di gioco con Amazon GameLift Servers code

Funzionamento di Amazon GameLift Servers

Questo argomento descrive come Amazon GameLift Servers gestisce l'hosting dedicato per i server di gioco multiplayer e come renderli disponibili ai giocatori. Descrive come funzionano le funzionalità principali.

Hosting di server di gioco

ConAmazon GameLift Servers, puoi ospitare i tuoi server di gioco in diversi modi: gestiti Amazon GameLift Servers e Amazon GameLift Servers ovunque. Amazon GameLift Servers FleetIQ
Per ulteriori informazioni su Amazon GameLift ServersFleetIQ, vedi Cos'è Amazon GameLift ServersFleetIQ?

È possibile progettare un parco istanze in base alle esigenze di gioco. Per ulteriori informazioni sulla progettazione di una flotta, consultaPersonalizza il tuo Amazon GameLift Servers EC2 flotte gestite.

Amazon GameLift Servers gestito

Con managedAmazon GameLift Servers, puoi ospitare i tuoi server di gioco su risorse informatiche Amazon GameLift Servers virtuali, chiamate istanze. Configura le tue risorse di hosting creando una flotta di istanze e distribuendole per far funzionare i tuoi server di gioco.

Amazon GameLift ServersOvunque

Con Amazon GameLift Servers Anywhere, puoi ospitare i tuoi server di gioco sui computer che gestisci. Configura le tue risorse di hosting creando una flotta Anywhere che faccia riferimento al tuo sistema di calcolo.

Alias del parco istanze

Un alias è una designazione che puoi trasferire tra flotte, il che lo rende un modo conveniente per avere un'ubicazione generica del parco veicoli. Puoi usare un alias per cambiare client di gioco da una flotta all'altra senza cambiare client di gioco. Puoi anche creare un alias di terminale che indichi il contenuto.

Sessioni di gioco in corso

Dopo aver distribuito la build del server di gioco su una flotta e aver Amazon GameLift Servers avviato i processi del server di gioco su ogni istanza, la flotta può ospitare sessioni di gioco. Amazon GameLift Serversavvia nuove sessioni di gioco quando il servizio client di gioco invia una richiesta di posizionamento al servizio di backend o a. Amazon GameLift Servers

Posizionamento della sessione di gioco e algoritmo FleetIQ

Le code utilizzano l'FleetIQalgoritmo per selezionare un server di gioco disponibile per ospitare una nuova sessione di gioco. Il componente chiave per il posizionamento della sessione di gioco è la coda della sessione di Amazon GameLift Servers gioco. Assegni alla coda delle sessioni di gioco un elenco di flotte, che determina dove la coda può collocare le sessioni di gioco. Per ulteriori informazioni sulle code delle sessioni di gioco e su come progettarle per il gioco, consulta. Personalizza una coda di sessioni di gioco

Ottimizzazione del posizionamento delle sessioni di gioco con i beacon ping UDP

Utilizzando i beacon di ping Amazon GameLift Servers UDP, puoi calcolare la latenza di andata e ritorno per i pacchetti UDP tra giocatori e server di gioco in posizioni diverse per aiutarti a scegliere la posizione ottimale per una sessione di gioco. Per ulteriori informazioni sui beacon di ping UDP e su come utilizzarli per misurare la latenza, consulta. Fari di ping UDP

- Per i giochi che utilizzano le code delle sessioni di gioco per il posizionamento, la richiesta di posizionamento può includere dati sulla latenza, che la coda utilizza automaticamente per dare priorità alle posizioni di posizionamento. È possibile personalizzare ulteriormente l'assegnazione delle priorità e impostare politiche che includono limiti di latenza. Consultare Dai priorità al posizionamento delle sessioni di gioco.
- Per i giochi che non utilizzano code di sessione di gioco ma hanno flotte con più sedi, puoi valutare i dati di latenza e scegliere la migliore posizione disponibile prima di effettuare una richiesta di sessione di gioco. Amazon GameLift Servers Consulta le sezioni Ottieni e Crea sessioni di gioco in. Aggiungi Amazon GameLift Servers al tuo client di gioco
- Se lo utilizzi FlexMatch per il matchmaking, puoi impostare le regole di abbinamento per utilizzare i dati di latenza. Vedi Richiedi matchmaking per i giocatori e la sezione Regola di latenza relativa ai tipi di FlexMatch regole nella Guida per gli sviluppatori. Amazon GameLift Servers FlexMatch

Connessioni dei giocatori ai giochi

Come parte del processo di posizionamento della sessione di gioco, la coda o la sessione di gioco richiede al server di gioco selezionato di iniziare una nuova sessione di gioco. Il server di gioco risponde alla richiesta e segnala Amazon GameLift Servers quando è pronto ad accettare le connessioni dei giocatori. Amazon GameLift Serversquindi fornisce le informazioni di connessione al servizio di backend o al servizio client di gioco. I tuoi client di gioco utilizzano queste informazioni per connettersi direttamente alla sessione di gioco e iniziare il gioco.

Dimensionamento della capacità del parco istanze

Quando una flotta è attiva e pronta per ospitare sessioni di gioco, puoi modificarne la capacità per soddisfare la domanda dei giocatori. Ti consigliamo di trovare un equilibrio tra il fatto che tutti i giocatori entranti trovino rapidamente una partita e che spendano troppo in risorse che rimangono inutilizzate.

Amazon GameLift Serversoffre uno strumento di ridimensionamento automatico altamente efficace oppure è possibile impostare manualmente la capacità del parco veicoli. Per ulteriori informazioni, consulta Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers.

Dimensionamento automatico

Amazon GameLift Serversfornisce due metodi di ridimensionamento automatico:

- Scalabilità automatica basata sull'obiettivo

- Scalabilità automatica con policy basate su regole

Altre caratteristiche del dimensionamento

- Protezione delle sessioni di gioco: Amazon GameLift Servers impedisci di terminare le sessioni di gioco che ospitano giocatori attivi durante un evento a scala ridotta.
- Limiti di scalabilità: controlla l'utilizzo complessivo delle istanze impostando limiti minimi e massimi sul numero di istanze in un parco istanze.
- Sospensione della scalabilità automatica: sospendi la scalabilità automatica a livello di sede della flotta senza modificare o eliminare le politiche di scalabilità automatica.
- Metriche di scalabilità: tieni traccia della cronologia della capacità e degli eventi di scalabilità di una flotta.

Monitoraggio di Amazon GameLift Servers

Quando disponi di flotte attive e funzionanti, Amazon GameLift Servers raccoglie una serie di informazioni per aiutarti a monitorare le prestazioni dei server di gioco distribuiti. Puoi utilizzare queste informazioni per ottimizzare l'uso delle risorse, risolvere problemi e ottenere informazioni dettagliate sul modo in cui i giocatori sono attivi nei tuoi giochi. Amazon GameLift Serversraccoglie quanto segue:

- Dettagli sulla flotta, sulla posizione, sulla sessione di gioco e sulla sessione del giocatore
- Parametri di utilizzo
- Stato del processo del server
- Registri delle sessioni di gioco

Per ulteriori informazioni sul monitoraggio inAmazon GameLift Servers, vedereMonitoraggio Amazon GameLift Servers.

Utilizzo di altre AWS risorse

I server e le applicazioni di gioco possono comunicare con altre AWS risorse. Ad esempio, potresti utilizzare un set di servizi Web per l'autenticazione dei giocatori o i social network. Affinché i server di gioco possano accedere alle AWS risorse Account AWS gestite da te, consenti esplicitamente l'accesso Amazon GameLift Servers alle tue AWS risorse.

Amazon GameLift Servers fornisce un paio di opzioni per la gestione di questo tipo di accesso. Per ulteriori informazioni, consulta Comunica con altre AWS risorse delle tue flotte.

Come funzionano i contenitori in Amazon GameLift Servers

Amazon GameLift Serversle flotte di container sono progettate per offrirti flessibilità nel modo in cui distribuisci e ridimensiona le tue applicazioni containerizzate. Utilizza Amazon Elastic Container Service (Amazon ECS) per gestire la distribuzione e l'esecuzione delle attività per le tue flotte. Amazon GameLift Servers Questo argomento descrive gli elementi strutturali di base per far funzionare i container su una flotta Amazon GameLift Servers gestita, illustra le architetture comuni e delinea alcuni concetti fondamentali.

Accelera l'onboarding con questi strumenti per container gestiti:

- Lo starter kit per container semplifica l'integrazione e la configurazione della flotta. Aggiunge funzionalità essenziali per la gestione delle sessioni di gioco al server di gioco e utilizza modelli preconfigurati per creare una flotta di container e una pipeline di distribuzione automatizzata per il server di gioco. Dopo l'implementazione, utilizza la Amazon GameLift Servers console e gli strumenti API per monitorare le prestazioni della flotta, gestire le sessioni di gioco e analizzare le metriche.
- Per gli sviluppatori di Unreal Engine o Unity, usa i Amazon GameLift Serversplugin per integrare il tuo server di gioco e creare una flotta di container dall'interno dell'ambiente di sviluppo del tuo motore di gioco. I flussi di lavoro guidati del plug-in ti aiutano a creare una soluzione semplice e veloce con l'hosting basato su cloud utilizzando contenitori gestiti. Quindi sfrutta queste basi per creare una soluzione di hosting personalizzata per il tuo gioco.

Componenti della flotta di container

Parco istanze

Una flotta di container è una raccolta di EC2 istanze Amazon per ospitare i tuoi server di gioco containerizzati. Queste istanze sono gestite da Amazon GameLift Servers te per tuo conto. Quando crei una flotta, configuri il modo in cui l'architettura del container con il software del server di gioco viene distribuita a ciascuna istanza della flotta. Puoi creare una flotta di container con istanze in una o più località geografiche. Puoi utilizzare strumenti di Amazon GameLift Servers

scalabilità per scalare automaticamente la capacità di una flotta di container di ospitare sessioni di gioco e giocatori.

Istanza

Un' EC2 istanza Amazon è il server virtuale che fornisce capacità di elaborazione per l'hosting di giochi. ConAmazon GameLift Servers, puoi scegliere tra una vasta gamma di tipi di istanze. Ogni tipo di istanza offre una combinazione diversa di CPU, memoria, storage e capacità di rete.

Quando crei una flotta di container, Amazon GameLift Servers distribuisci i container in base al tipo di istanza scelto e alla configurazione della flotta. Ogni istanza della flotta distribuita è identica e esegue il software del server di gioco containerizzato nello stesso modo. Il numero di istanze in un parco istanze determina le dimensioni del parco istanze e la capacità di hosting dei giochi.

Gruppo di contenitori

Amazon GameLift Serversutilizza il concetto di gruppo di contenitori per descrivere e gestire un insieme di contenitori. Un gruppo di contenitori è simile a un contenitore «task» o «pod». All'interno di ciascun gruppo di contenitori, è possibile definire il comportamento dei contenitori, impostare le dipendenze e condividere le risorse di CPU e memoria disponibili.

Ogni istanza della flotta può avere i seguenti tipi di gruppi di contenitori:

- Un gruppo di contenitori di server di gioco gestisce i contenitori che eseguono l'applicazione del server di gioco e il software di supporto. Una flotta di container deve avere uno di questo tipo di gruppo di container per ospitare sessioni di gioco e giocatori. Un gruppo di container di server di gioco può essere replicato su un'istanza della flotta. Il numero di repliche di gruppi di server di gioco per istanza del parco istanze dipende dai requisiti di elaborazione del software e dalle risorse di calcolo disponibili sull'istanza.
- Un gruppo di container per istanza, opzionale, ti dà la possibilità di eseguire software aggiuntivo su ogni istanza del parco istanze. Sono utili per eseguire servizi in background o programmi di utilità, ad esempio per il monitoraggio. Il software del server di gioco non dipende direttamente dai processi di un gruppo di istanze. In ogni istanza del parco veicoli viene distribuita una sola copia di un gruppo di container per istanza.

Ogni gruppo di container in una flotta di container ha un container designato come «essenziale». Un container essenziale determina il ciclo di vita di un gruppo di container. Se il contenitore essenziale si guasta, l'intero gruppo di contenitori si riavvia.

Container

Il contenitore è l'elemento più basilare di un'architettura basata su contenitori. Include un'immagine del contenitore con eseguibili software e file dipendenti. Definisci un contenitore per configurare il modo in cui il software viene eseguito e con cui interagisce. Amazon GameLift Servers

Amazon GameLift Serversdefinisce due tipi di contenitori:

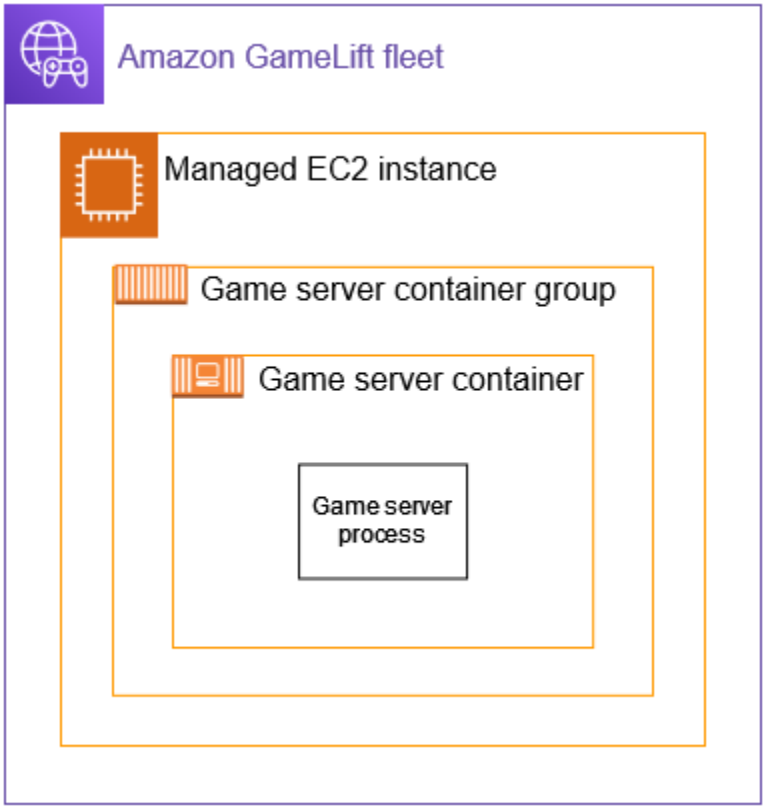
- Un contenitore per server di gioco include tutto ciò di cui hai bisogno per eseguire i processi del server di gioco e ospitare sessioni di gioco per i giocatori. Include la build del server di gioco e il software dipendente. Definisci un contenitore di server di gioco per il gruppo di contenitori di server di gioco di una flotta. Il contenitore del server di gioco viene automaticamente considerato essenziale per il gruppo di contenitori.
- Un contenitore di supporto esegue software aggiuntivo per supportare il server di gioco. È simile al concetto di contenitore «sidecar». Ti dà la possibilità di eseguire e scalare il software di supporto insieme ai server di gioco, gestendolo però come contenitori separati. In un gruppo di contenitori di server di gioco, puoi definire zero o più contenitori di supporto. In un gruppo di contenitori per istanza, tutti i contenitori sono contenitori di supporto. Qualsiasi contenitore di supporto può essere considerato essenziale.

Calcolo

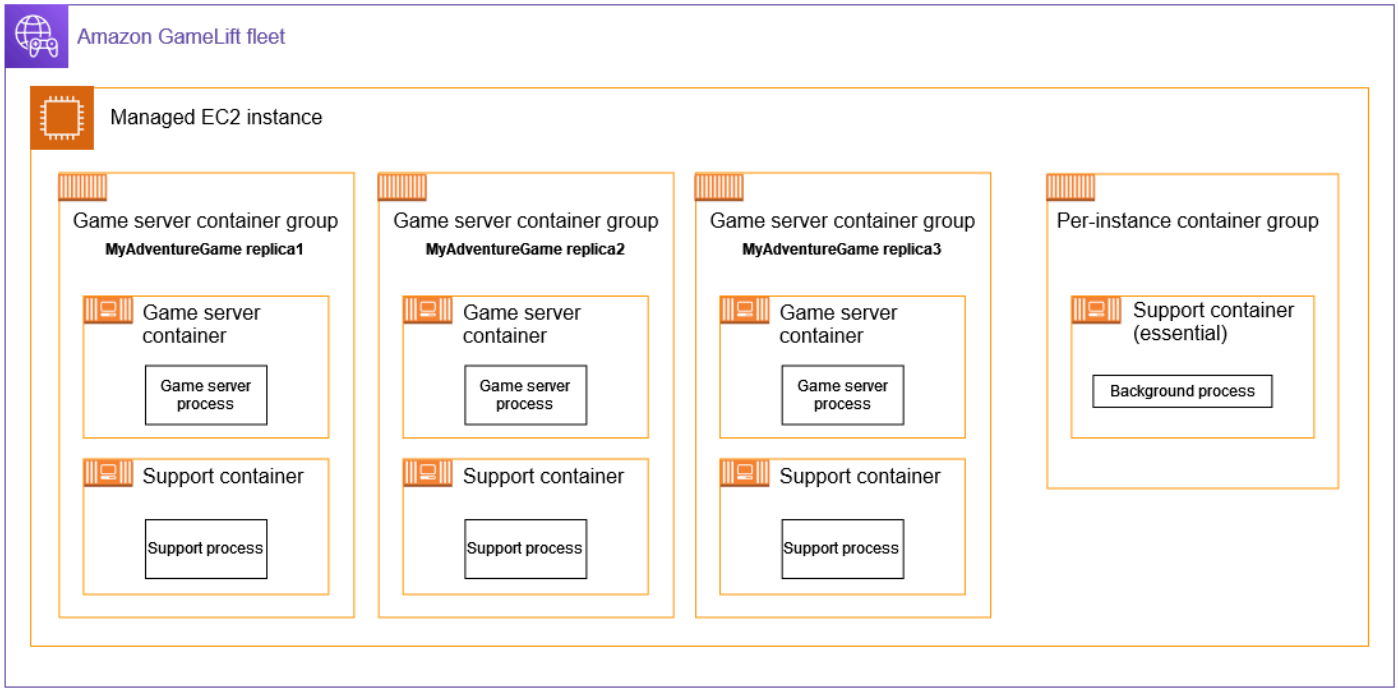
Un calcolo rappresenta una copia di un gruppo di contenitori di server di gioco su un'istanza della flotta.

Architetture comuni

Il diagramma seguente illustra la struttura più semplice della flotta di container. In questa struttura, ogni istanza della flotta mantiene una copia del gruppo di contenitori del server di gioco. Il gruppo di contenitori ha un unico contenitore di server di gioco che esegue un processo del server di gioco. In questo esempio, la flotta di container è configurata per collocare una copia del gruppo di contenitori del server di gioco per istanza. Con questa architettura, ogni istanza esegue un processo del server di gioco.



Questo secondo diagramma illustra un'architettura di flotte di container più complessa. In questa struttura, la flotta dispone sia di un gruppo di container di server di gioco che di un gruppo di container per istanza. Il gruppo di contenitori di server di gioco ha contenitori separati per il processo del server di gioco e un processo di supporto. Il parco macchine è configurato per posizionare tre copie del gruppo di contenitori del server di gioco su ciascuna istanza del parco macchine. Il gruppo di contenitori per istanza non viene mai replicato. In questo esempio, la flotta di container è configurata per collocare tre copie del gruppo di contenitori del server di gioco per istanza. Con questa architettura, ogni istanza esegue tre processi del server di gioco.



Caratteristiche principali

Questa sezione riassume come Amazon GameLift Servers implementa alcuni concetti di base sui container. Per istruzioni su come lavorare con le flotte di container, consulta gli argomenti pertinenti di questa guida.

Aggiornamenti attivi della flotta

I container gestiti forniscono un supporto avanzato per aiutarti a gestire il ciclo di vita del software ospitato e dell'architettura dei container. Puoi aggiornare le definizioni dei container, incluse le immagini dei container, e implementare modifiche alle flotte esistenti. Questa funzionalità semplifica e velocizza l'iterazione delle modifiche ai container durante lo sviluppo. Fornisce inoltre funzionalità che consentono di creare, distribuire e tenere traccia degli aggiornamenti delle versioni del software nel tempo. Queste funzionalità includono:

- Gestisci gli aggiornamenti e il controllo delle versioni delle definizioni dei gruppi di contenitori. È possibile aggiornare quasi tutte le proprietà di una definizione di gruppo di contenitori, incluse le immagini dei contenitori e le impostazioni di configurazione. Ogni volta che aggiorni un contenitore, assegna Amazon GameLift Servers automaticamente un numero di versione all'aggiornamento e, per impostazione predefinita, mantiene tutte le versioni. È possibile accedere a qualsiasi versione specifica ed eliminare le versioni desiderate. Quando si crea una flotta di container, è possibile specificare la definizione del gruppo di container e la versione da distribuire.

- Aggiorna le flotte di container esistenti con nuove definizioni di gruppi di contenitori e impostazioni di configurazione. Puoi distribuire gli aggiornamenti dei container alle flotte già distribuite nelle istanze del parco veicoli. Puoi monitorare lo stato delle distribuzioni degli aggiornamenti in ogni sede del parco veicoli utilizzando AWS Management Console o l' AWS SDK e la CLI.
- Configura il modo in cui desideri che gli aggiornamenti del parco macchine vengano distribuiti in un parco macchine attivo.
  - Protezione delle sessioni di gioco. Scegli di proteggere le istanze della flotta con sessioni di gioco attive fino al termine delle sessioni di gioco (distribuzione sicura). Oppure scegli di sostituire le istanze della flotta indipendentemente dall'attività della sessione di gioco (distribuzione non sicura). Utilizza implementazioni non sicure durante le fasi di sviluppo e test per ridurre i tempi di implementazione.
  - Percentuale minima di salute. Specificate la percentuale di attività integre che desiderate mantenere durante la distribuzione. Questa funzionalità ti consente di decidere quante istanze del parco istanze subire l'impatto durante una distribuzione. Un valore basso dà priorità alla velocità di implementazione, mentre un valore alto assicura che la disponibilità dei server di gioco rimanga elevata per tutta la durata dell'implementazione.
  - Strategia di fallimento dell'implementazione. Decidi quali azioni intraprendere se una distribuzione fallisce. Un errore di distribuzione significa che alcuni contenitori aggiornati non hanno superato i controlli di stato e sono considerati compromessi. È possibile impostare le distribuzioni per ripristinare automaticamente tutte le istanze del parco istanze allo stato di distribuzione precedente. In alternativa, puoi scegliere di mantenere alcune delle istanze del parco istanze danneggiate da utilizzare nel debug.

La possibilità di aggiornare le flotte attive è molto utile quando desideri implementare un aggiornamento al software del tuo server di gioco. Dopo aver creato una nuova immagine del container per il server di gioco, la sua implementazione è un processo in due fasi: in primo luogo, aggiorna la definizione del gruppo di container con la nuova immagine e, in secondo luogo, aggiorna la flotta di container. Amazon GameLift Serversgestisce tutte le altre attività in base alle esigenze.

Imballaggio in contenitori

Quando si sviluppa la struttura dei container per l'implementazione in una flotta di container, un obiettivo comune è ottimizzare l'uso della potenza di calcolo disponibile. Per raggiungere questo obiettivo, devi raggruppare il maggior numero possibile di gruppi di container di server di gioco in ciascuna istanza della flotta.

Amazon GameLift Serversti aiuta a farlo calcolando il numero massimo di gruppi di contenitori di server di gioco per istanza, in base alle seguenti informazioni:

- Il tipo di istanza del parco istanze e le relative risorse di vCPU e memoria.
- I requisiti di vCPU e memoria per tutti i contenitori nel gruppo di contenitori del server di gioco.

  I requisiti di vCPU e memoria per tutti i contenitori in un gruppo di contenitori per istanza, se presente.

Quando si crea una flotta di container, è possibile utilizzare il massimo calcolato o specificare il numero desiderato. Come procedura ottimale, pianifica di sperimentare con il software del server di gioco containerizzato per determinare i requisiti di risorse per prestazioni ottimali del server di gioco.

Dimensionamento della capacità

La capacità della flotta misura il numero di sessioni di gioco che una flotta può ospitare contemporaneamente. Puoi anche misurare la capacità in base al numero di giocatori simultanei che una flotta può supportare. Per aumentare o diminuire la capacità di hosting di una flotta, aggiungi o rimuovi istanze della flotta.

Una flotta di container è configurata per eseguire un numero specifico di processi simultanei di server di gioco su ciascuna istanza della flotta. (Puoi calcolarlo in base (1) ai gruppi di contenitori dei server di gioco per istanza e (2) al numero di processi dei server di gioco eseguiti in ciascun gruppo di contenitori.) Il numero di server di gioco simultanei per istanza indica l'impatto dell'aggiunta o della rimozione di ciascuna istanza del parco istanze. Ad esempio, se la tua flotta di container esegue 1 processo di server di gioco in ogni gruppo di container di server di gioco e ogni istanza della flotta contiene 100 gruppi di contenitori di server di gioco, aumenti o diminuisci la capacità della tua flotta di ospitare sessioni di gioco simultanee con incrementi di 100. Se ogni sessione di gioco ha 10 slot per giocatori, puoi aumentare o diminuire la capacità della flotta di ospitare giocatori con incrementi di 1000.

Con le flotte di container, puoi utilizzare uno qualsiasi dei metodi di scalabilità della capacità forniti da. Amazon GameLift Servers Ciò include:

- Imposta manualmente la capacità della flotta impostando il numero di istanze della flotta desiderato.
- Imposta il ridimensionamento automatico scegliendo come target un buffer desiderato di istanze disponibili (tracciamento del target). Questo metodo mantiene automaticamente una certa quantità

di risorse di hosting inattive in modo che i giocatori in arrivo possano iniziare rapidamente a giocare. Man mano che la domanda dei giocatori aumenta o diminuisce, la dimensione di questo buffer viene continuamente adattata.

- Imposta il ridimensionamento automatico con regole di ridimensionamento personalizzate (funzionalità avanzata). Questo metodo ti consente di scalare in base alle metriche della flotta che scegli.

Connessioni di gioco client/server

Con le flotte Amazon GameLift Servers gestite, i client di gioco si connettono direttamente ai server di gioco ospitati sul cloud. Quando un client di gioco chiede di partecipare a una partita, Amazon GameLift Servers trova una sessione di gioco e fornisce le informazioni di connessione (IP e porta) al client di gioco. Puoi controllare l'accesso esterno alle istanze della flotta aprendo determinati intervalli di porte (autorizzazioni in entrata) per la flotta. Le autorizzazioni in entrata determinano quali porte sono aperte al traffico in entrata. Puoi chiudere rapidamente tutte le porte, limitarle a poche o aprire tutte le porte.

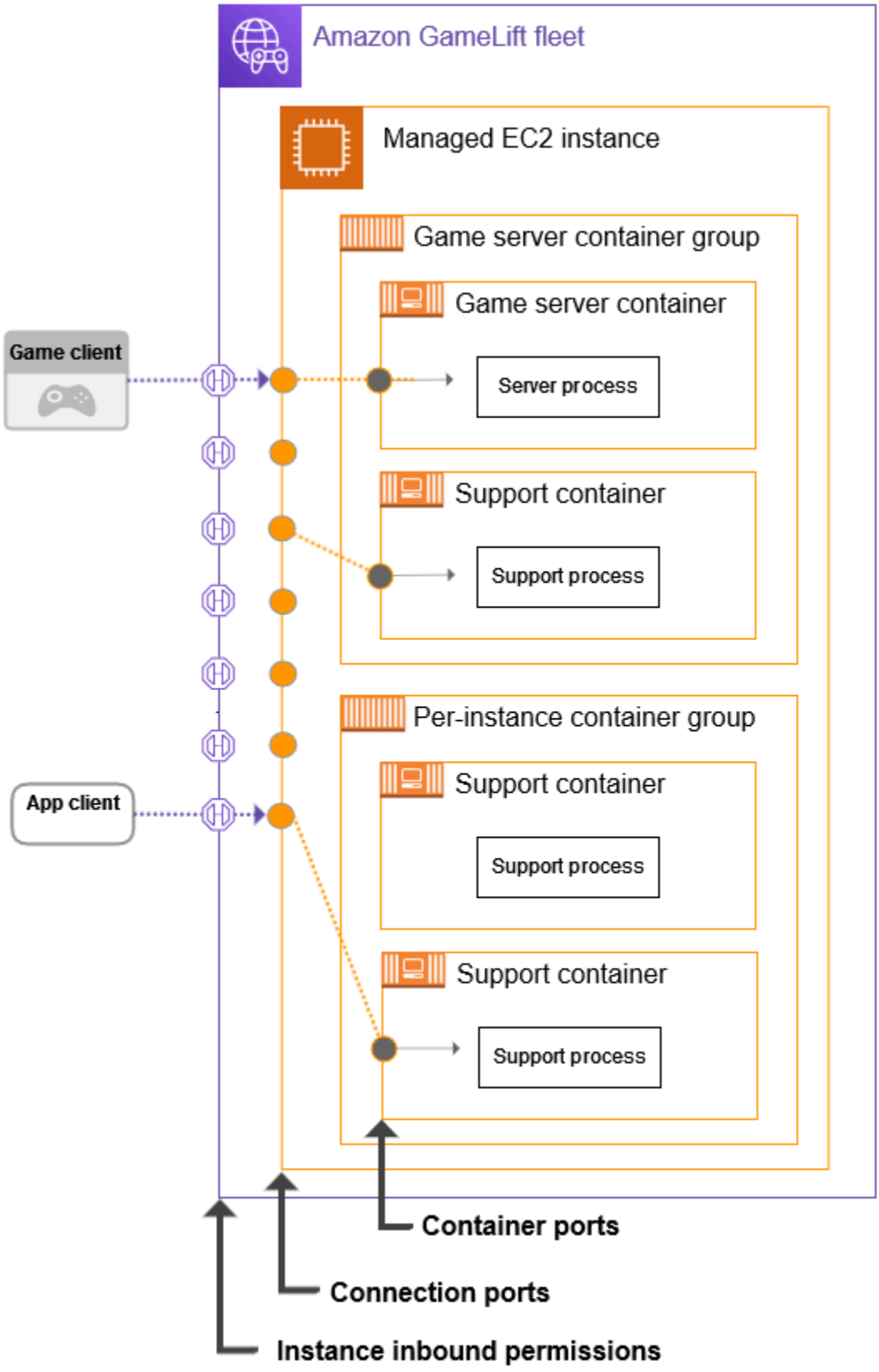
Le flotte di container gestite richiedono un'impostazione aggiuntiva che consenta l'accesso ai processi in esecuzione in un container. Quando si crea una definizione di contenitore, si specifica un set di porte, una per ogni processo che richiede una connessione. Questo include:

- Tutti i processi del server di gioco che verranno eseguiti contemporaneamente nel contenitore del server di gioco. Tutti i processi del server di gioco devono consentire ai client di gioco di connettersi per partecipare a una sessione di gioco.
- Qualsiasi processo in un contenitore di supporto a cui deve connettersi una fonte esterna. Ad esempio, puoi connetterti in remoto a un'app di test.

Quando configuri le impostazioni della porta container rivolta verso l'interno, le Amazon GameLift Servers utilizza per calcolare le autorizzazioni in entrata rivolte all'esterno a cui possono connettersi i client di gioco e altre applicazioni. Amazon GameLift Serversgestisce anche la mappatura tra le autorizzazioni in entrata e le singole porte container che consente ai giocatori di accedere a una sessione di gioco in un contenitore. Questa mappatura interna fornisce un livello di sicurezza proteggendo i server di gioco dall'accesso diretto alle porte dei container. Hai la possibilità di personalizzare le impostazioni delle porte rivolte verso l'esterno di una flotta in base alle esigenze. Per ulteriori informazioni sull'impostazione manuale dei porti della flotta di container, consulta. Configurare le connessioni di rete

Puoi modificare le impostazioni dei porti di una flotta di container in qualsiasi momento. Questa modifica richiede l'implementazione di un aggiornamento della flotta.

Il diagramma seguente illustra il ruolo delle connessioni portuali all'interno di una flotta di container. Come illustrato, si impostano le porte sui singoli container e Amazon GameLift Servers si utilizzano queste informazioni per configurare un numero sufficiente di porte sull'istanza della flotta da mappare a ciascun porto container. Sia le autorizzazioni in ingresso delle istanze rivolte verso l'esterno che le porte di connessione rivolte verso l'interno vengono calcolate in base al tuo parco macchine, a meno che tu non scelga di impostarle manualmente. Amazon GameLift Servers


Amazon GameLift fleet
Managed EC2 instance
Game server container group
Game server container
Server process
Support container
Support process
Per-instance container group
Support container
Support process
Support container
Support process
Game client
App client
Container ports
Connection ports
Instance inbound permissions

Registrazione dei container

Nelle flotte di container gestite, i flussi di output standard (e di errore standard) vengono acquisiti per tutti i container. Ciò include i registri delle sessioni di gioco del server di gioco. Puoi configurare una flotta di container per utilizzare una delle diverse opzioni per gestire i flussi di output:

- Salva l'output del contenitore come flusso di CloudWatch log di Amazon. Ogni flusso di log fa riferimento all'ID della flotta e al container. Se si sceglie questa opzione di registrazione per il parco veicoli, si specifica un gruppo di CloudWatch log che organizza tutti i flussi di log del parco veicoli. È quindi possibile utilizzare CloudWatch le funzionalità per cercare e analizzare i dati di registro in base alle esigenze.
- Salva l'output del contenitore in un bucket di storage Amazon Simple Storage Service (Amazon S3). Puoi visualizzare, condividere o scaricare il contenuto secondo necessità.
- Disattiva la registrazione. In questo scenario, l'output del contenitore non viene salvato.

Amazon GameLift Serversinvia i dati di registro dalle flotte di container gestite ai CloudWatch servizi Amazon S3 del tuo account. AWS Per visualizzare i tuoi dati, utilizza questo AWS Management Console o altri strumenti accedendo al tuo AWS account e lavorando con i singoli servizi. Estendi l'accesso limitato Amazon GameLift Servers per intraprendere queste azioni creando un ruolo di servizio per le tue flotte di container.

Puoi modificare la configurazione di registrazione di una flotta di container in qualsiasi momento. Questa modifica richiede l'implementazione di un aggiornamento della flotta.

Le flotte di container e l'agente Amazon GameLift Servers

Un'architettura di container di uso comune esegue un singolo processo in ogni contenitore. In una flotta di Amazon GameLift Servers container, il gruppo di contenitori di server di gioco dispone di un contenitore di server di gioco, che esegue un processo del server di gioco. Con questa architettura, Amazon GameLift Servers gestisce il ciclo di vita del singolo processo del server di gioco in ogni gruppo di contenitori di server di gioco su un'istanza della flotta.

Se scegli di creare un'architettura container che esegua più processi di server di gioco in ogni gruppo di contenitori di server di gioco, hai bisogno di un modo per gestire il ciclo di vita di tutti i processi. Ciò include attività come l'avvio, l'arresto e la sostituzione dei processi in base alle esigenze, la gestione del numero desiderato di processi da eseguire contemporaneamente e la gestione degli stati di errore.

È possibile scegliere di utilizzare l'Amazon GameLift Serversagente per queste attività. Per una flotta di container, l'agente implementa istruzioni di runtime che specificano quali eseguibili eseguire (e quanti), forniscono i parametri di avvio e stabiliscono regole per l'attivazione del server di gioco. Ad esempio, le istruzioni di runtime potrebbero indicare all'agente di mantenere dieci processi del server di gioco per l'uso in produzione e un processo del server di gioco con parametri di avvio speciali per i test.

Per utilizzare l'agente con le flotte di container, aggiungete l'agente all'immagine del contenitore e includete un set di istruzioni di runtime. Per ulteriori informazioni sull'agente, consultaCollabora con l'Amazon GameLift Serversagente.

Come i giocatori si connettono ai giochi

Una sessione di gioco è un'istanza del gioco in esecuzione su Amazon GameLift Servers. Per giocare, un giocatore può trovare e partecipare a una sessione di gioco esistente oppure creare una nuova sessione di gioco e unirsi a essa. Un giocatore si unisce creando una sessione per la sessione di gioco. Se la sessione di gioco è aperta ai giocatori, Amazon GameLift Servers riserva uno slot per il giocatore e fornisce informazioni sulla connessione. Il giocatore può quindi connettersi alla sessione gioco e richiedere lo slot prenotato.

Per informazioni dettagliate sulla creazione e la gestione di sessioni di gioco e sessioni di gioco con server di gioco personalizzati, consultaAggiungi Amazon GameLift Servers al tuo client di gioco. Per informazioni su come connettere i giocatori a Amazon GameLift ServersRealtime, consultaIntegrazione di un client di gioco per Amazon GameLift ServersRealtime.

Amazon GameLift Serversoffre diverse funzionalità relative alle sessioni di gioco e ai giocatori.

Organizza sessioni di gioco utilizzando le migliori risorse disponibili in più sedi

> Scelta fra varie opzioni quando si configura in che modo Amazon GameLift Servers seleziona le risorse per l'hosting di nuove sessioni di gioco. Se gestisci flotte in più località, puoi progettare code per le sessioni di gioco che inseriscono nuove sessioni di gioco su qualsiasi flotta indipendentemente dalla posizione. Usa i ping beacon UDP per calcolare i dati di latenza, che la coda utilizza automaticamente per dare priorità alle posizioni di posizionamento.

Controlla l'accesso dei giocatori alle sessioni di gioco

> Configura le sessioni di gioco per consentire o negare le richieste di iscrizione di nuovi giocatori, indipendentemente dal numero di giocatori connessi.

Usa dati di gioco e giocatori personalizzati

Aggiungi dati personalizzati agli oggetti della sessione di gioco e agli oggetti della sessione del giocatore. Amazon GameLift Serverspassa i dati della sessione di gioco a un server di gioco quando inizia una nuova sessione di gioco. Amazon GameLift Serverspassa i dati della sessione del giocatore al server di gioco quando un giocatore si connette alla sessione di gioco.

Filtra e ordina le sessioni di gioco disponibili

Usa la ricerca e l'ordinamento delle sessioni per trovare la migliore corrispondenza possibile per un potenziale giocatore, oppure consenti al giocatore di scegliere da un elenco di sessioni di gioco disponibili. Usa la ricerca e l'ordinamento delle sessioni per trovare le sessioni di gioco in base alle caratteristiche della sessione. Puoi anche utilizzare la funzione di ricerca e ordinamento in base ai tuoi dati personalizzati del gioco.

Tieni traccia dei dati di utilizzo del gioco e dei giocatori

Archivia automaticamente i log delle sessioni di gioco completate. Puoi configurare l'archiviazione dei log durante l'Amazon GameLift Serversintegrazione nei tuoi server di gioco. Per ulteriori informazioni, consulta Registrazione dei messaggi del server Amazon GameLift Servers.

Usa la Amazon GameLift Servers console per visualizzare informazioni dettagliate sulle sessioni di gioco, inclusi i metadati delle sessioni, le impostazioni e i dati delle sessioni dei giocatori. Per ulteriori informazioni, consultare Sessioni di gioco e di gioco nella Amazon GameLift Servers console e Metriche.

Amazon GameLift Serverssedi di assistenza

Amazon GameLift Serversle funzionalità sono disponibili su più Regioni AWS Local Zones. Puoi progettare una soluzione di hosting che colloca i server di gioco esattamente dove si trovano i giocatori.

AWS Località supportate

La tabella seguente è un elenco delle Regioni AWS Local Zones che supportano Amazon GameLift Servers le risorse. Indica i tipi di risorse che è possibile creare in ogni posizione.

Ubicazione geografica	Codice di ubicazione	Regione d'origine per le flotte gestite (sede unica)	Area geografica per le flotte gestite (con più sedi)	Localizzazione remota per flotte gestite (più sedi)	Flotta ovunque	Coda della sessione di gioco	FlexMatch matchmaker e set di regole
US East (N. Virginia)	us-east-1	Sì	Sì	Sì	Sì	Sì	Sì
Stati Uniti orientali (Ohio)	us-east-2	Sì		Sì	Sì	Sì	
US West (N. California)	us-west-1	Sì		Sì	Sì	Sì	
US West (Oregon)	us-west-2	Sì	Sì	Sì	Sì	Sì	Sì
Africa (Cape Town)	af-south-1			Sì			
Asia Pacifico (Tailandia)	ap-southeast-7			Sì			
Asia Pacifico (Hong Kong)	ap-east-1			Sì			
Asia Pacifico (Malesia)	ap-southeast-5			Sì			
Asia Pacific (Mumbai)	ap-south-1	Sì		Sì	Sì	Sì	

Ubicazione geografica	Codice di ubicazione	Regione d'origine per le flotte gestite (sede unica)	Area geografica per le flotte gestite (con più sedi)	Localizzazione remota per flotte gestite (più sedi)	Flotta ovunque	Coda della sessione di gioco	FlexMatch matchmaker e set di regole
Asia Pacifico (Osaka-Local)	ap-northeast-3			Sì			
Asia Pacifico (Seul)	ap-northeast-2	Sì	Sì	Sì	Sì	Sì	Sì
Asia Pacifico (Singapore)	ap-southeast-1	Sì		Sì	Sì	Sì	
Asia Pacifico (Sydney)	ap-southeast-2	Sì	Sì	Sì	Sì	Sì	Sì
Asia Pacifico (Tokyo)	ap-northeast-1	Sì	Sì	Sì	Sì	Sì	Sì
Canada (Central)	ca-central-1	Sì		Sì	Sì	Sì	
Europe (Frankfurt)	eu-central-1	Sì	Sì	Sì	Sì	Sì	Sì
Europa (Irlanda)	eu-west-1	Sì	Sì	Sì	Sì	Sì	Sì
Europe (London)	eu-west-2	Sì		Sì	Sì	Sì	

Ubicazione geografica	Codice di ubicazione	Regione d'origine per le flotte gestite (sede unica)	Area geografica per le flotte gestite (con più sedi)	Localizzazione remota per flotte gestite (più sedi)	Flotta ovunque	Coda della sessione di gioco	FlexMatch matchmaker e set di regole
Europa (Milano)	eu-south-1			Sì			
Europe (Paris)	eu-west-3			Sì			
Europa (Stoccolma)	eu-north-1			Sì			
Medio Oriente (Bahrein)	me-south-1			Sì			
Sud America (São Paulo)	sa-east-1	Sì		Sì	Sì	Sì	
Zona locale di Atlanta	Stati Uniti est-1-atl-1			Sì			
Zona locale di Chicago	us-east-1-chi-1			Sì			
Zona locale di Dallas	us-east-1-dfw-1			Sì			
Zona locale di Denver	us-west-2-den-1			Sì			

Ubicazione geografica	Codice di ubicazione	Regione d'origine per le flotte gestite (sede unica)	Area geografica per le flotte gestite (con più sedi)	Localizzazione remota per flotte gestite (più sedi)	Flotta ovunque	Coda della sessione di gioco	FlexMatch matchmaker e set di regole
Zona locale di Houston	usa-est-1-iah-1			Sì			
Zona locale di Kansas City	us-east-1-mci-1			Sì			
Zona locale di Los Angeles	us-west-2-lax-1			Sì			
Zona locale di Phoenix	us-west-2-phx-1			Sì			
Zona locale di Lagos, Nigeria	af-south-1-los-1			Sì			

Note

Non tutti Regioni AWS sono abilitati di default per un. Account AWS Se desideri una flotta con più sedi con istanze in queste regioni, devi abilitarle. Per ulteriori informazioni sulle regioni che non sono abilitate per impostazione predefinita e su come abilitarle, consulta Gestione Regioni AWS in. Riferimenti generali di AWS Le flotte create prima del 28 febbraio 2022 non sono interessate.

Inoltre, devi aggiornare la politica Amazon GameLift Servers dell'amministratore per consentire l'azione`ec2:DescribeRegions`. Per un esempio di politica con le regioni

che non sono abilitate per impostazione predefinita, consultaEsempi di autorizzazioni amministrative.

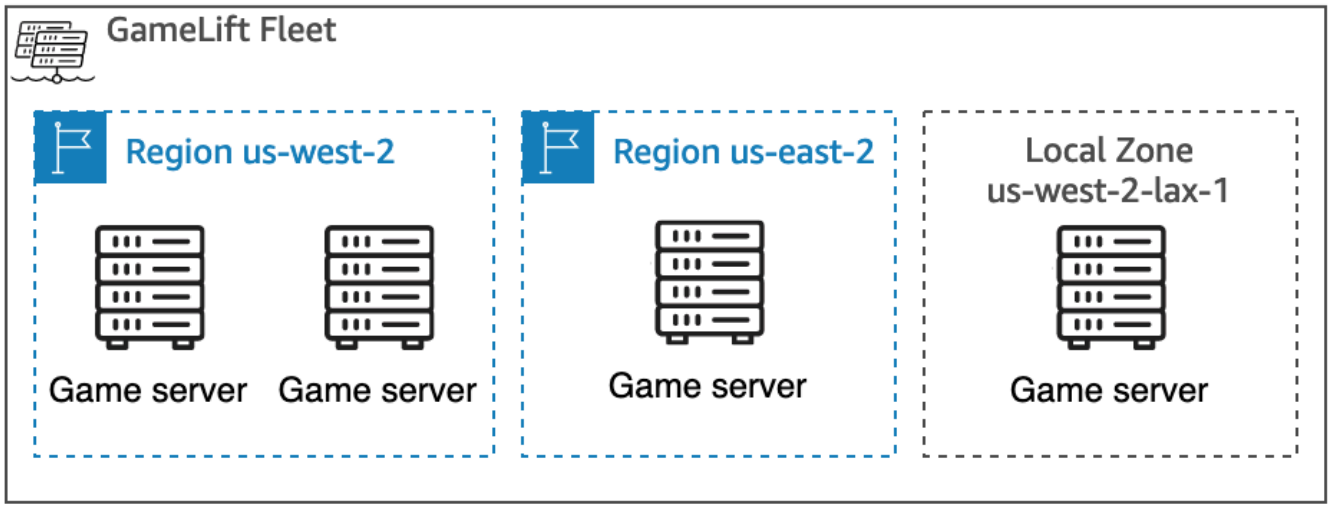
Sedi per l'hosting gestito

Amazon GameLift Serversl'hosting gestito distribuisce flotte di risorse dei server di gioco. Ogni flotta viene creata in una Regione AWS, che è la regione di origine della flotta. La regione di origine di una flotta è indicata nell'Amazon Resource Number (ARN) della flotta.

Puoi implementare una flotta in una sola regione, con risorse di hosting solo nella regione d'origine. In alternativa, puoi implementare una flotta con più sedi, con risorse di hosting in più località geografiche. Una flotta con più sedi ha un'area geografica e una o più sedi remote. Quando gestisci la capacità di hosting per una flotta con più sedi, puoi impostare la capacità per ciascuna sede individualmente.

Le località remote per una flotta con più sedi possono essere altre Regioni AWS o Local Zones. Una zona locale è un'estensione di un Regione AWS. Consente di collocare le risorse di elaborazione più vicine agli utenti per fornire un gameplay a bassa latenza. Per ulteriori informazioni, consulta AWS Zone locali. Il codice di posizione di una zona locale è il codice regionale principale seguito da un identificatore di posizione fisico. Ad esempio, il codice per la zona locale di Los Angeles è`us-west-2-lax-1`.

Il diagramma seguente illustra una flotta con più sedi con risorse in due Regioni AWS e una zona locale. La regione di origine della flotta è`us-west-2`, e ha due sedi remote: `us-east-2` Regione e `us-west-2-lax-1` Zona locale.



Oltre alle risorse della flotta, l'hosting gestito con Amazon GameLift Servers può utilizzare anche i seguenti tipi di risorse. Queste risorse vengono create in uno specifico Regione AWS che supporti il tipo di risorsa.

- Build: si tratta di un server di gioco costruito per essere ospitato con una EC2 flotta gestita. Crea una risorsa di costruzione nella stessa regione della flotta in cui verrà impiegata.
- Script: si tratta di uno script di configurazione con Amazon GameLift Servers Realtime cui ospitare un gioco. Crea una risorsa di script nella stessa regione della flotta in cui verrà distribuita.
- Definizione del gruppo di container e immagine del contenitore: si tratta di una configurazione per l'esecuzione di container su una flotta di container gestita. Identifica una o più immagini di container con un software da implementare nella flotta di container. Crea una definizione di gruppo di contenitori e tutte le immagini dei container (archiviate in un repository Amazon Elastic Container Registry) nella stessa regione della flotta in cui verranno distribuiti.
- Coda delle sessioni di gioco: questa risorsa elabora le richieste di sessioni di gioco e avvia nuove sessioni di gioco. L'elaborazione avviene nel luogo in Regione AWS cui si trova la coda. Per ridurre la latenza nel processo di posizionamento delle sessioni di gioco, crea una coda geograficamente vicina ai giocatori che la useranno.

Posizioni per qualsiasi luogo Amazon GameLift Servers

Una flotta Amazon GameLift Servers Anywhere è una raccolta di hardware di hosting che fornisci. Gestisci tutte le attività sulle tue risorse di hosting, inclusa la distribuzione del software del server di gioco, l'aggiornamento e l'avvio dei processi del server. Crei una flotta Anywhere per connettere il Amazon GameLift Servers servizio con le tue risorse di hosting autogestite. Amazon GameLift Serversgestisce il posizionamento delle sessioni di gioco: elabora le richieste di iscrizione dei giocatori, individua le risorse di hosting disponibili, avvia nuove sessioni di gioco e fornisce ai client di gioco le informazioni di connessione. Puoi creare una flotta Anywhere in uno qualsiasi dei sistemi che li supportano. Regioni AWS

Puoi aggiungere istanze di hardware di hosting a una flotta Anywhere registrandola. A ogni istanza registrata deve essere associata una posizione personalizzata. Le posizioni personalizzate non sono correlate a Regioni AWS o alle Local Zones. Vengono utilizzate per rappresentare la posizione fisica dell'hardware.

Per ulteriori informazioni sulla creazione di una flotta Anywhere e sul test dell'integrazione con i server di gioco, consulta Crea un Amazon GameLift Servers Flotta ovunque eConfigura i test locali con Amazon GameLift Servers Anywhere.

Sedi per Amazon GameLift ServersFlexMatch

FlexMatchle risorse vengono utilizzate per elaborare le richieste dei giocatori per il matchmaking. Includono una risorsa di configurazione del matchmaking e una risorsa per il set di regole. L'elaborazione avviene nel luogo in Regione AWS cui si trovano le FlexMatch risorse. Per ridurre la latenza nel processo di matchmaking, crea le risorse geograficamente vicino ai giocatori che le useranno. Una configurazione di matchmaking e il set di regole che utilizza devono trovarsi nella stessa configurazione. Regione AWS Puoi creare FlexMatch risorse in qualsiasi sistema Regioni AWS che le supporti.

Per ulteriori informazioni sulla configurazione della tua soluzione di hosting, consulta la guida FlexMatch per gli Amazon GameLift ServersFlexMatch sviluppatori.

Amazon GameLift Serversin Cina

Quando si utilizzano Amazon GameLift Servers risorse nella regione Cina (Pechino), gestita da Sinnet, o nella regione Cina (Ningxia), gestita da NWCD, è necessario disporre di un account separato (AWS Cina). Tieni presente che alcune funzionalità non sono disponibili nelle regioni della Cina. Per ulteriori informazioni sull'utilizzo Amazon GameLift Servers in queste regioni, consulta le seguenti risorse:

- Amazon Web Services in Cina
- Amazon GameLift Servers(Guida introduttiva ad Amazon Web Services in Cina)

Prezzi per Amazon GameLift Servers

Prezzi per Amazon GameLift Servers varia in base al tipo di servizio e alle funzionalità utilizzate. Per una discussione completa sui prezzi, inclusi gli scenari di esempio, consulta Amazon GameLift Servers Prezzi.

È possibile utilizzare… Amazon GameLift Servers senza incorrere in costi con il piano AWS gratuito. Puoi utilizzare i vantaggi del piano gratuito se sei AWS cliente da meno di 12 mesi e rispetti i limiti di utilizzo del piano gratuito. Per ulteriori informazioni, consulta Piano gratuito AWS.

Le basi di costo per ciascuna Amazon GameLift Servers i servizi sono i seguenti:

- Amazon GameLift Servers hosting gestito: il pagamento si basa (1) sull'utilizzo orario dell'istanza per ospitare sessioni di gioco e (2) sul trasferimento dei dati per il traffico tra client di gioco e server

di gioco ospitati. I costi di utilizzo delle istanze variano in base al tipo/dimensione dell'istanza Regione AWS, al sistema operativo e al fatto che si tratti di un'istanza Spot o On-Demand.

- Amazon GameLift Servers FlexMatch indipendente: paghi in base (1) al numero di richieste di matchmaking effettuate dai giocatori e (2) al tempo di elaborazione necessario per valutare le richieste di partite e proporre le partite.
- Amazon GameLift Servers Hosting ovunque: paghi in base (1) al numero di sessioni di gioco Amazon GameLift Servers posiziona sui computer Anywhere e (2) il numero di minuti di connessione ai processi server.
- Amazon GameLift Servers FleetiQ: paghi un addebito per EC2 ogni istanza in un gruppo di server di gioco. L'addebito deriva dal prezzo orario dell'istanza e si aggiunge al costo di utilizzo. EC2 I costi di utilizzo delle istanze variano in base al tipo/dimensione dell'istanza Regione AWS, al sistema operativo e al fatto che si tratti di un'istanza Spot o On-Demand.

Argomenti

Genera Amazon GameLift Servers stime dei prezzi

Con Calcolatore dei prezzi AWS, puoi creare una stima dei prezzi per Amazon GameLift Servers. Non è necessaria una conoscenza approfondita Account AWS o approfondita AWS per utilizzare la calcolatrice.

Calcolatore dei prezzi AWS la calcolatrice ti guida attraverso le decisioni che influiscono sui costi del servizio per darti un'idea di quanto Amazon GameLift Servers potrebbe costare per il tuo progetto di gioco. Se non sei ancora sicuro di come intendi utilizzarlo Amazon GameLift Servers, quindi utilizza i valori predefiniti per generare una stima. Quando si pianifica l'utilizzo della produzione, la calcolatrice può aiutarvi a testare potenziali scenari e generare stime più accurate.

È possibile utilizzare Calcolatore dei prezzi AWS per generare stime per quanto segue Amazon GameLift Servers opzioni di hosting:

- Stima Amazon GameLift Servers hosting gestito

Stima Amazon GameLift Servers hosting gestito

Questa opzione fornisce una stima dei costi per l'hosting dei tuoi giochi su Amazon GameLift Servers server gestiti, compresi i costi per l'utilizzo delle istanze del server e il trasferimento dei dati. Con Amazon GameLift Servers hosting gestito, non ci sono costi aggiuntivi per FlexMatch matchmaking.

Se ospiti o prevedi di ospitare server di gioco in più di una AWS regione o su più di un tipo di istanza, crea una stima per ogni regione e tipo di istanza.

Amazon GameLift Servers Istanze

Questa sezione ti aiuta a stimare il tipo e il numero di risorse di calcolo necessarie per ospitare sessioni di gioco per i tuoi giocatori. Amazon GameLift Servers utilizza istanze Amazon Elastic Compute Cloud (Amazon EC2) per gestire i server di gioco. In Amazon GameLift Servers, distribuisci una flotta di istanze con un tipo di istanza e un sistema operativo specifici. Se disponi o prevedi di avere più flotte, crea una stima per ciascuna flotta.

Per iniziare, apri Configure Amazon GameLift Servers pagina di Calcolatore dei prezzi AWS. Aggiungi una descrizione, scegli una regione, quindi scegli Stima Amazon GameLift Servers hosting (Instance + Data Transfer Out). In Amazon GameLift Servers istanze, completa i seguenti campi:

- Numero massimo di giocatori concorrenti (picco di potenza della CCU)

  Questo è il numero massimo di giocatori che possono connettersi contemporaneamente ai tuoi server di gioco. Questo campo indica la capacità di hosting Amazon GameLift Servers deve soddisfare il picco di domanda dei giocatori. Inserisci il numero massimo giornaliero di giocatori che prevedi di ospitare utilizzando le istanze nella AWS regione prescelta.

  Ad esempio, se desideri consentire a 1.000 giocatori di connettersi al tuo gioco contemporaneamente, mantieni il valore predefinito di**`1000`**.
- CCU media all'ora come percentuale della CCU giornaliera di picco

  Questo è il numero medio di giocatori simultanei all'ora in un periodo di 24 ore. Utilizziamo questo valore per stimare la quantità di capacità di hosting sostenuta che Amazon GameLift Servers deve essere mantenuto per i tuoi giocatori. Se non sei sicuro del valore percentuale da utilizzare, mantieni il valore predefinito della **`50`** percentuale. Per i giochi con una domanda stabile da parte dei giocatori, ti consigliamo di inserire un valore **`70`** percentuale.

  Ad esempio, se il gioco ha una CCU oraria media di 6.000 e una CCU di picco di 10.000, inserisci il valore percentuale. **`60`**

- Sessioni di gioco per esempio

  Questo è il numero di sessioni di gioco che ciascuna istanza del server di gioco può ospitare contemporaneamente. I fattori che possono influire su questo numero includono il fabbisogno di risorse del server di gioco, il numero di giocatori da ospitare in ogni sessione di gioco e le aspettative in termini di prestazioni dei giocatori. Se conosci il numero di sessioni di gioco simultanee per il tuo gioco, inserisci quel valore. In alternativa, mantieni il valore predefinito di. **20**

- Giocatori per sessione di gioco

  Questo è il numero medio di giocatori che si connettono a una sessione di gioco, come definito nella progettazione del gioco. Se utilizzi modalità di gioco con un numero diverso di giocatori, calcola un numero medio di giocatori per sessione di gioco nell'arco dell'intera partita. Il valore predefinito è **8**.

- Buffer di inattività dell'istanza%

  Questa è la percentuale di capacità di hosting inutilizzata da mantenere come riserva per gestire i picchi improvvisi della domanda da parte dei giocatori. La dimensione del buffer è una percentuale del numero totale di istanze in un parco istanze. Il valore predefinito è **10** percentuale.

  Ad esempio, con un buffer di inattività del 20%, un parco istanze che supporta giocatori con 100 istanze attive mantiene 20 istanze inattive.

- Istanza Spot%

  Amazon GameLift Servers le flotte possono utilizzare una combinazione di istanze On-Demand e istanze Spot. Mentre le istanze on demand offrono una disponibilità più affidabile, le istanze Spot offrono un'alternativa altamente efficiente in termini di costi. Ti consigliamo di utilizzare una combinazione per ottimizzare sia il risparmio sui costi che la disponibilità. Per informazioni su come Amazon GameLift Servers utilizza istanze Spot, vediIstanze On-Demand e istanze Spot.

  Per questo campo, inserisci la percentuale di istanze Spot da gestire in un parco istanze. Consigliamo una percentuale di istanze Spot compresa tra il 50 e l'85 percento. Il valore predefinito è **50** percentuale.

  Ad esempio, se distribuisci un parco istanze con 100 istanze e specifichi **40** la percentuale, Amazon GameLift Servers lavora per mantenere 60 istanze on demand e 40 istanze Spot.

- Tipo di istanza

Amazon GameLift Servers le flotte possono utilizzare una gamma di tipi di EC2 istanze Amazon che variano in termini di potenza di calcolo, memoria, archiviazione e capacità di rete. Quando configuri un Amazon GameLift Servers fleet, scegli il tipo di istanza più adatto alle esigenze del tuo gioco. Per informazioni sulla selezione di un tipo di istanza con Amazon GameLift Servers, consulta Scegli le risorse di elaborazione per una flotta gestita.

Se conosci il tipo di istanza che stai utilizzando o intendi utilizzare nel tuo Amazon GameLift Servers flotta, scegli quel tipo. Se non sei sicuro del tipo da scegliere, valuta la possibilità di scegliere c5.large. Si tratta di un tipo ad alta disponibilità con dimensioni e funzionalità medie.

- Sistema operativo

  Questo campo specifica il sistema operativo su cui girano i server di gioco, Linux o Windows. Il valore predefinito è Linux.

Trasferimento dati in uscita (DTO)

Questa sezione ti aiuta a stimare il costo del traffico tra i tuoi client di gioco e i server di gioco. Le tariffe per il trasferimento dei dati si applicano solo al traffico in uscita. Il trasferimento dei dati in entrata non ha costi.

Nella sezione Configura Amazon GameLift Servers pagina di Calcolatore dei prezzi AWS, espandi Data transfer out (DTO), quindi completa i seguenti campi:

- Tipo di stima DTO

  Puoi scegliere di stimare il DTO in uno dei due modi seguenti, a seconda di come tieni traccia del trasferimento dei dati durante il gioco.

  - Al mese (in GB): se registri il traffico mensile per i tuoi server di gioco, scegli questo tipo.
  - Per giocatore: se tieni traccia del trasferimento di dati per giocatore, scegli questo tipo. Questo è il tipo predefinito.

    Nel campo seguente, si stima il DTO per giocatore in base al numero di ore di gioco calcolato nella sezione precedente.
- DTO al mese (in GB)

  Se hai scelto il tipo di stima DTO mensile (in GB), inserisci l'utilizzo mensile stimato del DTO in GB per ogni istanza, per regione.

- DTO per giocatore

  Se hai scelto il tipo di stima del DTO per giocatore, inserisci l'utilizzo stimato del DTO per giocatore per giocatore in KB/sec. Il valore predefinito è **4**.

Quando hai finito di configurare il Amazon GameLift Servers stima del prezzo, scegli Aggiungi al mio preventivo. Per ulteriori informazioni sulla creazione e la gestione delle stime in Calcolatore dei prezzi AWS, consulta Creare una stima, configurare un servizio e aggiungere altri servizi nella Guida per l'Calcolatore dei prezzi AWS utente.

Gestisci i tuoi Amazon GameLift Servers costi di hosting

La AWS fattura riflette i costi di hosting dei giochi. Puoi visualizzare gli addebiti stimati per il mese corrente e gli addebiti finali per i mesi precedenti sulla console di fatturazione all'indirizzo https://console.aws.amazon.com/costmanagement/. Per ulteriori informazioni sugli strumenti e le risorse per aiutarti a gestire i AWS costi, consulta la Guida per l'AWS Billing utente. Questa guida può aiutarti a esaminare il consumo di risorse, stabilire l'utilizzo futuro e determinare le tue esigenze di scalabilità.

Considerate questi suggerimenti per aiutarvi a gestire i costi di Amazon GameLift Servers servizi.

Crea avvisi di fatturazione per monitorare l'utilizzo

Imposta un avviso di utilizzo del piano AWS gratuito per avvisarti quando l'utilizzo si avvicina o supera i limiti del piano gratuito per Amazon GameLift Servers e altro. Servizi AWSÈ possibile configurare gli avvisi in modo che intervengano in base ai livelli di utilizzo. Ad esempio, puoi impostare automaticamente il budget su zero quando raggiungi un limite del piano gratuito.

Puoi anche impostare avvisi di CloudWatch fatturazione di Amazon per ricevere notifiche quando l'utilizzo raggiunge soglie personalizzate.

Per ulteriori informazioni, consulta questi argomenti nella Guida per l'utente:AWS Billing

- Monitoraggio dell'utilizzo del piano AWS gratuito
- Preferenze relative agli avvisi di fatturazione

Tieni traccia dei costi per Amazon GameLift Servers parco istanze

Utilizza i tag di allocazione dei AWS costi per organizzare e tenere traccia dei costi di hosting dei giochi in base a Amazon GameLift Servers EC2 Flotte Amazon e altre risorse. Etichettando le flotte,

individualmente o per gruppi, puoi creare report di allocazione dei costi che classificano i costi in base al tag assegnato. Puoi utilizzare questo tipo di rapporto per identificare in che modo le flotte contribuiscono ai costi di hosting. Puoi anche utilizzare i tag per filtrare le viste in AWS Cost Explorer.

Per ulteriori informazioni, consulta i seguenti argomenti:

- Utilizzo AWS dei tag di allocazione dei costi,AWS Billing Guida per l'utente
- Analisi dei costi con AWS Cost Explorer, Guida per AWS Cost Management l'utente

Imposta a zero la capacità gestita della flotta

Le flotte gestite possono continuare a sostenere costi anche quando non vengono utilizzate per ospitare sessioni di gioco. Per evitare di incorrere in costi inutili, riduci a zero la flotta quando non viene utilizzata. Se utilizzi la scalabilità automatica, sospendi questa attività e imposta manualmente la capacità del parco veicoli.

Nozioni di base su Amazon GameLift Servers

Sfrutta queste risorse introduttive per saperne di più sul Amazon GameLift Servers servizio e su come iniziare a sviluppare una soluzione di hosting personalizzata per i tuoi giochi multiplayer basati su sessioni.

Prima di iniziare

- Creane uno Account AWS (o designane uno esistente) con cui utilizzarlo. Amazon GameLift Servers
- Configura gli utenti con le autorizzazioni Amazon GameLift Servers e i servizi correlati AWS .
- Seleziona un utente Regione AWS su cui lavorare. Per lo sviluppo, scegli una regione vicina alla tua posizione. Puoi cambiare regione in qualsiasi momento.

Configura un Account AWS

Opzioni di onboarding rapido

Prova questi strumenti di avvio rapido per ottenere una soluzione di hosting di base attiva e funzionante rapidamente con uno sviluppo semplificato. Questi strumenti sono ideali per la dimostrazione del concetto e la prototipazione, oppure possono essere utilizzati per creare ambienti di test per lo sviluppo rapido e iterativo di giochi. Dopo aver utilizzato questi strumenti per distribuire un server di gioco per l'hosting, puoi utilizzare la Amazon GameLift Servers console e gli strumenti API per monitorare le prestazioni della flotta, gestire le sessioni di gioco e analizzare le metriche.

- Game server wrapper Amazon GameLift Servers: questo strumento e il relativo tutorial di onboarding sono i modi più rapidi e semplici per ospitare ed eseguire sessioni di gioco sul server di gioco senza dover modificare il codice. Amazon GameLift Servers Readme, il wrapper per server di gioco, fornisce istruzioni da riga di comando per tutti i tipi di flotte e il tutorial illustra l'utilizzo della console e di un tipo di EC2 flotta gestita, in modo da poter iniziare subito. Segui le istruzioni per configurare la gestione delle sessioni di gioco e l'implementazione semplificata dei server di gioco.

  Quando sei pronto per creare una soluzione di hosting di giochi personalizzata, passa a una delle opzioni di sviluppo personalizzate con piena integrazione con l'SDK del server per. Amazon GameLift Servers Se il tuo gioco non necessita di una soluzione di hosting personalizzata, puoi

continuare a utilizzare il wrapper del server di gioco per distribuire e ospitare server di gioco in produzione.

- Amazon GameLift Serversplug-in per Unreal Engine o Unity: i plug-in offrono flussi di lavoro con interfaccia grafica e risorse di esempio per guidarti nei passaggi iniziali e implementare il tuo server di gioco con una soluzione di hosting di base. Utilizza il plug-in per configurare l'hosting con flotte Anywhere autogestite o implementare flotte o flotte di container gestite e basate sul cloud. EC2 Quando sei pronto a sviluppare una soluzione di hosting personalizzata, puoi sfruttare le soluzioni integrate nei plug-in.
- Starter kit per container Amazon GameLift Servers gestiti: questo kit semplifica le attività per integrare un server di gioco, preparare l'immagine di un container del server di gioco e implementare una flotta di container per l'hosting. Per quanto riguarda l'integrazione, il kit aggiunge funzionalità essenziali per la gestione delle sessioni di gioco al server di gioco. Il kit utilizza modelli preconfigurati per creare una flotta di container e una pipeline di distribuzione automatizzata per il server di gioco. Quando sei pronto per aggiungere funzionalità complete di gestione delle sessioni di gioco, segui una delle roadmap di sviluppo personalizzate per integrare l'SDK del server. Amazon GameLift Servers

Opzioni di sviluppo personalizzate

Segui una di queste roadmap di sviluppo per iniziare a creare una soluzione di hosting personalizzata completa per il tuo gioco. Le tabelle di marcia forniscono indicazioni dettagliate su come creare, testare e personalizzare ogni componente della tua soluzione di hosting.

- Roadmap di sviluppo per l'hosting con managed Amazon GameLift Servers EC2
- Roadmap di sviluppo per l'hosting con contenitori Amazon GameLift Servers gestiti
- Roadmap di sviluppo per l'hosting con Anywhere Amazon GameLift Servers
- Roadmap di sviluppo per l'hosting ibrido con Amazon GameLift Servers

Esempi di Amazon GameLift Servers

Se stai pensando di Amazon GameLift Servers utilizzarlo per gestire il tuo server di gioco personalizzato o sei interessato a sfruttarlo Amazon GameLift ServersRealtime, ti consigliamo di provare i seguenti esempi prima di utilizzare il servizio per il tuo gioco. L'esempio di server di gioco personalizzato ti offre un'esperienza con l'hosting di giochi sulla Amazon GameLift Servers console.

L'Amazon GameLift ServersRealtimeesempio mostra come preparare un gioco per l'hosting tramite Realtime server.

Esempio di server di gioco personalizzato

Questo esempio dimostra il processo di implementazione di un server di gioco di esempio su una EC2 flotta Amazon GameLift Servers gestita per l'hosting. Usa il client di gioco di esempio per connetterti a una sessione di gioco dal vivo. Puoi scoprire come utilizzare Amazon GameLift Servers .tools, tra cui la console e la AWS CLI, per monitorare le prestazioni e l'utilizzo dell'hosting della flotta.

L'esempio illustra i seguenti passaggi:

- Carica la build di esempio del server di gioco.
- Crea una flotta per eseguire la build del server di gioco.
- Scarica il client di gioco di esempio e usalo per connetterti a un server di gioco e partecipare a una sessione di gioco.
- Rivedi le metriche della flotta e delle sessioni di gioco.

Avvia più client di gioco e gioca per generare dati di hosting. Usa la Amazon GameLift Servers console per visualizzare le risorse di hosting, tenere traccia delle metriche ed esplorare le opzioni per scalare la capacità di hosting della flotta.

Per iniziare, accedi alla Amazon GameLift Servers console. Nella barra di navigazione a sinistra, vai su Risorse, Prova un gioco di esempio.

Amazon GameLift ServersRealtimeesempio

Questo esempio è un tutorial completo che spiega come implementare un gioco multiplayer di esempio, Mega Frog Race, con. Amazon GameLift Servers Realtime Il tutorial spiega come integrare il client di gioco con l'RealtimeSDK e implementare una soluzione di hosting completa con Realtime server su flotte gestite. EC2

Per un tutorial pratico, consulta Creazione di server per giochi multigiocatore per dispositivi mobili con solo poche righe JavaScript sul blog for Games. AWS Per il codice sorgente di Mega Frog Race, consulta il repository. GitHub

Il codice sorgente include le seguenti parti:

- Client di gioco: codice sorgente per il client di gioco C++, creato in Unity. Il client di gioco ottiene informazioni sulla connessione della sessione di gioco, si connette al server e scambia aggiornamenti con altri giocatori.
- Servizio di backend: codice sorgente per una AWS Lambda funzione che gestisce le chiamate dirette all'API del servizio perAmazon GameLift Servers.
- Realtimescript: un file di script sorgente che configura una flotta di Realtime server per il gioco. Questo script include la configurazione minima richiesta per ogni Realtime server per comunicare Amazon GameLift Servers e ospitare sessioni di gioco.

Dopo aver configurato il gioco di esempio per l'hosting, usalo come punto di partenza per sperimentare altre Amazon GameLift Servers funzionalità comeFlexMatch.

Configura un Account AWS

Per iniziare a utilizzareAmazon GameLift Servers, crea e configura il tuo Account AWS. Non ci sono costi per creare un Account AWS. Questa sezione illustra come creare il tuo account, configurare gli utenti e configurare le autorizzazioni.

Argomenti

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

Quando ti iscrivi a Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Imposta le autorizzazioni utente per Amazon GameLift Servers

Crea utenti aggiuntivi o estendi le autorizzazioni di accesso agli utenti esistenti in base alle esigenze delle tue Amazon GameLift Servers risorse. Come best practice (best practice di sicurezza in IAM), applica le autorizzazioni con privilegi minimi per tutti gli utenti. Per indicazioni sulla sintassi delle autorizzazioni, consulta. Esempi di autorizzazioni IAM per Amazon GameLift Servers

Utilizza le seguenti istruzioni per impostare le autorizzazioni utente in base a come gestisci gli utenti del tuo account. AWS

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in AWS IAM Identity Center:

Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center .

- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Create a role for a third-party identity provider (federation) della Guida per l'utente IAM.

- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella pagina Create a role for an IAM user della Guida per l'utente IAM.
  - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

Quando lavori con utenti IAM, come best practice, assegna sempre le autorizzazioni ai ruoli o ai gruppi di utenti, non ai singoli utenti.

Configura l'accesso programmatico per gli utenti

Gli utenti necessitano dell'accesso programmatico se desiderano interagire con l' AWS esterno di. AWS Management Console Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Utilizza credenziali temporane e per firmare le richieste programmatiche a AWS CLI,, AWS SDKs o. AWS APIs	Segui le istruzioni per l'interfa ccia che desideri utilizzare. • Per la AWS CLI, vedere Configurazione dell'uso AWS IAM Identity Center nella AWS CLI Guida per l'utente.AWS Command Line Interface

Quale utente necessita dell'accesso programmatico?	Per	Come
		• Per AWS SDKs gli strumenti e AWS APIs, consulta l'autenticazione di IAM Identity Center nella Guida di riferimento AWS SDKs and Tools.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	Seguendo le istruzioni riportate in Utilizzo delle credenziali temporanee con le AWS risorse nella Guida per l'utente IAM.
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare richieste programmatiche a AWS CLI,, AWS SDKs o. AWS APIs	Segui le istruzioni per l'interfa ccia che desideri utilizzare. • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente.AWS Command Line Interface • Per gli strumenti AWS SDKs e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimento agli strumenti e agli AWS SDKs strumenti. • Per AWS APIs, consulta la sezione Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

Se utilizzi le chiavi di accesso, consulta Best practice per la gestione delle chiavi di AWS accesso.

Configura l'accesso programmatico per il tuo gioco

La maggior parte dei giochi utilizza servizi di backend per comunicare Amazon GameLift Servers tramite. AWS SDKs Utilizza un servizio di backend (che funge da client di gioco) per richiedere sessioni di gioco, inserire giocatori in giochi e altre attività. Questi servizi richiedono un accesso programmatico e credenziali di sicurezza per autenticare le chiamate all'API del servizio. Amazon GameLift Servers

InfattiAmazon GameLift Servers, gestisci questo accesso creando un utente giocatore in AWS Identity and Access Management (IAM). Gestisci le autorizzazioni utente del giocatore tramite una delle seguenti opzioni:

- Crea un ruolo IAM con le autorizzazioni dell'utente giocatore e consenti all'utente giocatore di assumere il ruolo quando necessario. Il servizio di backend deve includere il codice per assumere questo ruolo prima di effettuare richieste a. Amazon GameLift Servers In conformità con le migliori pratiche di sicurezza, i ruoli forniscono un accesso limitato e temporaneo. Puoi utilizzare i ruoli per carichi di lavoro in esecuzione su AWS risorse (ruoli IAM) o all'esterno di AWS (IAM Roles Anywhere).
- Crea un gruppo di utenti IAM con autorizzazioni utente giocatore e aggiungi il tuo utente giocatore al gruppo. Questa opzione fornisce all'utente giocatore credenziali a lungo termine, che il servizio di backend deve archiviare e utilizzare per comunicare. Amazon GameLift Servers

Per la sintassi della politica di autorizzazione, vedi. Esempi di autorizzazioni utente per i giocatori

Per ulteriori informazioni sulla gestione delle autorizzazioni per l'uso da parte di un carico di lavoro, consulta IAM Identities: Temporary credentials in IAM.

Esempi di autorizzazioni IAM per Amazon GameLift Servers

Usa la sintassi in questi esempi per impostare le autorizzazioni AWS Identity and Access Management (IAM) per gli utenti che devono accedere alle Amazon GameLift Servers risorse. Per ulteriori informazioni sulla gestione delle autorizzazioni degli utenti, vedere. Imposta le autorizzazioni utente per Amazon GameLift Servers Quando gestisci le autorizzazioni per utenti esterni all'IAM Identity Center, come best practice, assegna sempre le autorizzazioni ai ruoli o ai gruppi di utenti IAM, non ai singoli utenti.

Se la utilizzi Amazon GameLift Servers FleetIQ come soluzione autonoma, consulta Configurare il tuo modulo. Account AWS Amazon GameLift Servers FleetIQ

Esempi di autorizzazioni amministrative

Questi esempi forniscono a un amministratore di hosting o a uno sviluppatore un accesso mirato alla gestione delle risorse di hosting dei Amazon GameLift Servers giochi.

Example Sintassi per le autorizzazioni Amazon GameLift Servers complete alle risorse di accesso

L'esempio seguente estende l'accesso completo a tutte le Amazon GameLift Servers risorse.

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

Example Sintassi per le autorizzazioni Amazon GameLift Servers delle risorse con supporto per le regioni che non sono abilitate per impostazione predefinita

L'esempio seguente estende l'accesso a tutte le Amazon GameLift Servers risorse e le AWS regioni che non sono abilitate per impostazione predefinita. Per ulteriori informazioni sulle regioni che non sono abilitate per impostazione predefinita e su come attivarle, vedi Gestione Regioni AWS in Riferimenti generali di AWS.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeRegions",
      "gamelift:*"
    ],
    "Resource": "*"
  }
}
```

Example Sintassi della Amazon GameLift Servers risorsa per accedere alle immagini dei contenitori in Amazon ECR

L'esempio seguente estende l'accesso alle azioni Amazon GameLift Servers di Amazon Elastic Container Registry (Amazon ECR) necessarie agli utenti quando lavorano con flotte di container gestite.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "ecr:DescribeImages",
      "ecr:BatchGetImage",
      "ecr:GetDownloadUrlForLayer"
    ],
    "Resource": "*"
  }
}
```

Example Sintassi delle risorse e delle Amazon GameLift Servers autorizzazioni **`PassRole`**

L'esempio seguente estende l'accesso a tutte le Amazon GameLift Servers risorse e consente a un utente di passare un ruolo di servizio IAM a. Amazon GameLift Servers Un ruolo di servizio offre una capacità Amazon GameLift Servers limitata di accedere ad altre risorse e servizi per conto dell'utente, come descritto inConfigurare un ruolo di servizio IAM per Amazon GameLift Servers. Ad esempio, quando si risponde a una `CreateBuild` richiesta, è Amazon GameLift Servers necessario accedere ai file di build in un bucket Amazon S3. Per ulteriori informazioni sull'`PassRole`azione, consulta IAM: Pass an IAM role to a specific AWS service nella IAM User Guide.

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

```
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "gamelift.amazonaws.com"
        }
      }
    }
  ]
}
```

Esempi di autorizzazioni utente per i giocatori

Questi esempi consentono a un servizio di backend o a un'altra entità di effettuare chiamate API all'Amazon GameLift ServersAPI. Coprono gli scenari comuni per la gestione delle sessioni di gioco, delle sessioni dei giocatori e del matchmaking. Per ulteriori dettagli, consulta Configura l'accesso programmatico per il tuo gioco.

Example Sintassi per i permessi di posizionamento delle sessioni di gioco

L'esempio seguente estende l'accesso alle code di posizionamento delle sessioni di gioco Amazon GameLift Servers APIs che utilizzano le code di posizionamento delle sessioni di gioco per creare sessioni di gioco e gestire le sessioni dei giocatori.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "PlayerPermissionsForGameSessionPlacements",
    "Effect": "Allow",
    "Action": [
      "gamelift:StartGameSessionPlacement",
      "gamelift:DescribeGameSessionPlacement",
      "gamelift:StopGameSessionPlacement",
      "gamelift:CreatePlayerSession",
      "gamelift:CreatePlayerSessions",
      "gamelift:DescribeGameSessions"
    ],
    "Resource": "*"
  }
}
```

Example Sintassi per le autorizzazioni di matchmaking

L'esempio seguente estende l'accesso a coloro Amazon GameLift Servers APIs che gestiscono FlexMatch le attività di matchmaking. FlexMatchabbina i giocatori per sessioni di gioco nuove o esistenti e avvia il posizionamento delle sessioni di gioco per le partite ospitate su. Amazon GameLift Servers Per ulteriori informazioni suFlexMatch, vedi Cos'è Amazon GameLift ServersFlexMatch?

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "PlayerPermissionsForGameSessionMatchmaking",
    "Effect": "Allow",
    "Action": [
      "gamelift:StartMatchmaking",
      "gamelift:DescribeMatchmaking",
      "gamelift:StopMatchmaking",
      "gamelift:AcceptMatch",
      "gamelift:StartMatchBackfill",
      "gamelift:DescribeGameSessions"
    ],
    "Resource": "*"
  }
}
```

Example Sintassi per le autorizzazioni di posizionamento manuale delle sessioni di gioco

L'esempio seguente estende l'accesso a coloro Amazon GameLift Servers APIs che creano manualmente sessioni di gioco e sessioni di giocatori su flotte specifiche. Questo scenario supporta giochi che non utilizzano code di posizionamento, ad esempio giochi che consentono ai giocatori di partecipare scegliendo da un elenco di sessioni di gioco disponibili (il metodo "list-and-pick").

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "PlayerPermissionsForManualGameSessions",
    "Effect": "Allow",
    "Action": [
      "gamelift:CreateGameSession",
      "gamelift:DescribeGameSessions",
      "gamelift:SearchGameSessions",
      "gamelift:CreatePlayerSession",
      "gamelift:CreatePlayerSessions",
```

```
      "gamelift:DescribePlayerSessions"
    ],
    "Resource": "*"
  }
}
```

Configurare un ruolo di servizio IAM per Amazon GameLift Servers

Alcune Amazon GameLift Servers funzionalità richiedono l'estensione dell'accesso limitato ad altre AWS risorse di tua proprietà. Puoi farlo creando un ruolo AWS Identity and Access Management (IAM). Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. Un ruolo IAM è simile a un utente IAM in quanto è un' AWS identità con politiche di autorizzazione che determinano ciò che l'identità può e non può fare. AWS Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque. Inoltre, un ruolo non ha credenziali a lungo termine standard associate (password o chiavi di accesso). Tuttavia, quando assumi un ruolo, vengono fornite le credenziali di sicurezza provvisorie per la sessione del ruolo.

Questo argomento spiega come creare un ruolo da utilizzare con le flotte Amazon GameLift Servers gestite. Se lo utilizzi Amazon GameLift Servers FleetIQ per ottimizzare l'hosting di giochi sulle tue istanze Amazon Elastic Compute Cloud (Amazon EC2), consulta Configurare il tuo Account AWS. Amazon GameLift Servers FleetIQ

Nella procedura seguente, crea un ruolo con una politica di autorizzazioni personalizzata e una politica di fiducia che Amazon GameLift Servers consenta di assumere il ruolo.

Crea un ruolo di servizio IAM per una flotta Amazon GameLift Servers gestita EC2

Fase 1: Creare una politica di autorizzazioni.

Utilizza le istruzioni e gli esempi in questa pagina per creare una politica di autorizzazioni personalizzata per il tipo di Amazon GameLift Servers flotta con cui lavori.

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Immettere o incollare un documento di policy JSON. Per dettagli sul linguaggio della policy IAM, consulta la Documentazione di riferimento delle policy JSON IAM.
6. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy nella Guida per l'utente di IAM.

7. (Facoltativo) Quando crei o modifichi una policy in AWS Management Console, puoi generare un modello di policy JSON o YAML da utilizzare nei modelli. AWS CloudFormation

   Per fare ciò, nell'editor delle politiche scegli Azioni, quindi scegli Genera modello. CloudFormation Per saperne di più AWS CloudFormation, consulta il riferimento al tipo di AWS Identity and Access Management risorsa nella Guida AWS CloudFormation per l'utente.
8. Una volta terminata l'aggiunta delle autorizzazioni alla policy, scegli Successivo.
9. Nella pagina Rivedi e crea, immettere un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
10. (Facoltativo) Aggiungere metadati alla policy collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consulta Tags for AWS Identity and Access Management resources in the IAM User Guide.
11. Seleziona Crea policy per salvare la nuova policy.

Fase 2: Crea un ruolo che Amazon GameLift Servers possa assumere.

Per creare un ruolo IAM

1. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
2. Nella pagina Seleziona entità attendibile, scegli l'opzione Politica di fiducia personalizzata. Questa selezione apre l'editor di criteri di fiducia personalizzati.
3. Sostituisci la sintassi JSON predefinita con la seguente, quindi scegli Avanti per continuare.

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

4. Nella pagina Aggiungi autorizzazioni, individua e seleziona la politica di autorizzazioni creata nel passaggio 1. Seleziona Successivo per continuare.
5. Nella pagina Nome, rivedi e crea, inserisci un nome di ruolo e una descrizione (opzionale) per il ruolo che stai creando. Controlla le entità Trust e le autorizzazioni aggiunte.
6. Scegli Crea ruolo per salvare il tuo nuovo ruolo.

Crea un ruolo IAM per i contenitori Amazon GameLift Servers gestiti

Se utilizzi container Amazon GameLift Servers gestiti, devi creare un ruolo di servizio IAM da utilizzare con una flotta di container. Questo ruolo concede le autorizzazioni limitate Amazon GameLift Servers necessarie per gestire le risorse della flotta di container e intraprendere azioni per tuo conto.

Per creare un ruolo IAM per una flotta di container

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.

3. Nella pagina Seleziona un'entità affidabile, scegli il AWS servizio e seleziona il caso d'uso "GameLift». Seleziona Next (Successivo).
4. In Aggiungi autorizzazioni, scegli la politica `GameLiftContainerFleetPolicy` gestita. Scegli Next (Successivo). AWS politiche gestite per Amazon GameLift ServersPer ulteriori informazioni su questa politica, consulta.
5. In Nome, rivedi e crea, inserisci il nome di un ruolo e scegli Crea ruolo per salvare il nuovo ruolo.

Sintassi della politica di autorizzazione

- Autorizzazioni per Amazon GameLift Servers assumere il ruolo di servizio

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

- Autorizzazioni per accedere alle AWS regioni che non sono abilitate per impostazione predefinita

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "gamelift.amazonaws.com",
          "gamelift.ap-east-1.amazonaws.com",
          "gamelift.me-south-1.amazonaws.com",
          "gamelift.af-south-1.amazonaws.com",
          "gamelift.eu-south-1.amazonaws.com"
        ]
```

```
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Ottieni strumenti Amazon GameLift Servers di sviluppo

Amazon GameLift Serversfornisce una serie di SDKs e altri strumenti per aiutarti a creare soluzioni di hosting di giochi per i tuoi giochi. SDKs Aggiungono funzionalità ai server di gioco, ai client di gioco e ai servizi di backend che consentono loro di interagire con il Amazon GameLift Servers servizio. Per le informazioni più recenti sulle versioni e sulla compatibilità dell'Amazon GameLift ServersSDK, consulta. Note di rilascio di Amazon GameLift Servers

Per i server di gioco

Integra e crea i tuoi server di gioco a 64 bit con il server SDK forAmazon GameLift Servers. Il server di gioco utilizza l'SDK del server per comunicare con il Amazon GameLift Servers servizio per la gestione delle sessioni di gioco, inclusi l'avvio, l'aggiornamento e l'interruzione delle sessioni di gioco. Per informazioni sull'integrazione dell'SDK del server nei tuoi progetti di gioco, consulta. Preparazione dei giochi per Amazon GameLift Servers

Supporto allo sviluppo

- Sistema operativo di sviluppo
  - Windows
  - Linux
- Linguaggi di programmazione

  Scarica l'Amazon GameLift ServersSDK. Per informazioni specifiche sulla versione e istruzioni di installazione, consultate i file readme inclusi in ogni pacchetto.

  - SDK per server C++
    - Riferimento all'SDK del server
    - Come effettuare l'integrazione
  - SDK per server C# (il supporto per .NET 4, .NET 6, .NET 8 varia in base alla versione, vedi) Versioni SDK

    - Riferimento all'SDK del server
    - Come effettuare l'integrazione
  - Vai al server SDK
    - Riferimento all'SDK del server
    - Come effettuare l'integrazione
- Supporto per motori di gioco

  Il plug-in completo Amazon GameLift Servers include flussi di lavoro dell'interfaccia utente e risorse di esempio, oltre a versioni integrate dell' AWS SDK e dell'SDK del server. I flussi di lavoro ti guidano attraverso come configurare e implementare il tuo server di gioco per l'hosting con flotte gestite, flotte di container gestite o EC2 flotte Anywhere autogestite. Se non ti servono i flussi di lavoro guidati, puoi scaricare solo l'SDK del server per il tuo motore di gioco dagli stessi repository. GitHub

  Se utilizzi un altro motore di gioco o un altro ambiente di sviluppo non supportato dal plug-in, scarica l'SDK del server per il tuo linguaggio di programmazione e aggiungilo al tuo progetto di gioco.

  Per informazioni specifiche sulla versione e istruzioni di installazione, consulta i file readme inclusi in ogni pacchetto.

  - Plugin per Unreal Engine: creato per l'uso con le versioni 5.5. Controlla i readme specifici della versione per il supporto di Unreal.
    - Guida ai plugin per Unreal Engine
    - Server C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers
  - Plugin per Unity: creato per l'uso con le versioni LTS di Unity Editor 6.0, 2022.3 o 2021.3. Supporta i profili.NET Framework e.NET Standard di Unity, con .NET Standard 2.1 e .NET 4.x. Controlla i file readme specifici della versione per il supporto di Unity.
    - Guida ai plugin per Unity
    - ???Riferimento all'SDK del server C#
  - Server SDK per Unreal
    - Riferimento all'SDK del server
    - Integrazione Amazon GameLift Servers in un progetto Unreal Engine
  - SDK per server per Unity
    - Riferimento all'SDK del server

- Integrazione Amazon GameLift Servers in un progetto Unity

Supporto in fase di esecuzione

Per una soluzione di hosting gestito, crea il tuo server di gioco in modo che funzioni su una delle seguenti immagini di macchine Amazon (AMIs). Amazon GameLift ServersVersioni AMIAmazon GameLift ServersPer ulteriori dettagli, consulta la pagina.

- Windows Server 2016
- Amazon Linux 2023
- Amazon Linux 2

> **Note**
>
> Amazon Linux 2 (AL2) terminerà il supporto il 30 giugno 2025. Scopri maggiori dettagli in Amazon Linux 2 FAQs. Per i server di gioco ospitati AL2 e che utilizzano il Amazon GameLift Servers server SDK 4.x, aggiorna prima la build del server di gioco al server SDK 5.x, quindi distribuiscila su 023 istanze. AL2 Consultare Migrazione al server SDK 5.x per Amazon GameLift Servers.

Strumenti aggiuntivi

Wrapper per server di gioco per Amazon GameLift Servers

Questo strumento ti aiuta a implementare un server di gioco per l'hosting con una serie di funzionalità di base per la gestione delle sessioni di gioco. Con questo strumento, non è necessario apportare modifiche al codice di gioco o integrare l'SDK del server per. Amazon GameLift Servers Usa il wrapper del server di gioco per impacchettare il tuo server di gioco e distribuirlo per l'hosting di giochi utilizzando una delle tre soluzioni di Amazon GameLift Servers hosting (Anywhere EC2, managed o managed container). Questo strumento è ideale per la valutazione precoce o la creazione di prototipi con il tuo gioco o con un gioco di esempio, poiché non supporta la personalizzazione del server di gioco. Se il tuo gioco non necessita di funzionalità personalizzate, puoi utilizzare il server di gioco con il wrapper del server di gioco per l'hosting di produzione.

Amazon GameLift ServersKit di strumenti

Il Amazon GameLift Servers Toolkit è una raccolta di script e altri strumenti che abbiamo sviluppato per aiutare gli sviluppatori con scenari e problemi comuni. I materiali del toolkit includono script, codice di esempio e file readme.

- Containers Starter Kit: utilizza questo strumento per semplificare le attività di configurazione delle build di server di gioco per l'hosting con contenitori gestiti. Amazon GameLift Servers Il kit integra le funzionalità essenziali di gestione delle sessioni di gioco in un server di gioco e utilizza modelli preconfigurati per creare una flotta di container e impostare una pipeline di distribuzione automatizzata per la creazione del server di gioco. Dopo l'implementazione, puoi monitorare le prestazioni della flotta, gestire le sessioni di gioco e analizzare le metriche utilizzando la Amazon GameLift Servers console e gli strumenti API. Il kit si integra con Amazon Simple Storage Service AWS CodeBuild per l'automazione delle build, per lo storage e AWS CloudFormation per l'implementazione dell'infrastruttura.
- Strumento di aggiornamento rapido della build: utilizza questo strumento per modificare una build di server di gioco già distribuita in una flotta gestita EC2 . Lo strumento è progettato per aiutarti a sostituire rapidamente i file di build del gioco senza dover configurare e creare nuove EC2 flotte ad ogni modifica. Puoi aggiornare singole istanze o tutte le istanze del parco istanze. Le opzioni ti consentono di sostituire file di build specifici o un'intera build e ti consentono di gestire il riavvio dei server di gioco dopo gli aggiornamenti.

Per i servizi client di gioco

Crea un servizio di backend a 64 bit per i tuoi client di gioco utilizzando l' AWS SDK, che include l'API di servizio per. Amazon GameLift Servers Il servizio di backend del tuo gioco gestisce le interazioni lato client con il Amazon GameLift Servers servizio, incluso l'avvio di nuove sessioni di gioco e l'aggiunta di giocatori alle partite.

Scarica l'SDK AWS

Per ulteriori informazioni sull'utilizzo dell' AWS SDK conAmazon GameLift Servers, consulta le seguenti risorse:

- Riferimento APIAmazon GameLift Servers
- Integrazione del servizio client
- Progetta un servizio di backend client

Per la Amazon GameLift Servers gestione delle risorse

Utilizza i seguenti strumenti per creare, aggiornare e monitorare le tue risorse di hosting Amazon GameLift Servers gestito.

- AWS Management Console— La AWS console è un'applicazione basata sul Web che fornisce l'accesso centralizzato a tutte le singole console di AWS servizio, tra cui. Amazon GameLift Servers Usa la console per creare o accedere a un AWS account e aprirla per utilizzare le Amazon GameLift Servers risorse di hosting dei giochi. Puoi configurare e distribuire flotte di hosting e altre risorse, visualizzare le metriche di utilizzo e prestazioni, tenere traccia delle risorse nella dashboard e molte altre attività. Vai alla console. Amazon GameLift Servers
- API di servizio per Amazon GameLift Servers: questa API ti offre l'accesso programmatico a tutte le tue Amazon GameLift Servers risorse. Fa parte dell' AWS SDK, che puoi scaricare per utilizzarlo con i linguaggi di programmazione più diffusi. Scarica l' AWS SDK.
- AWS interfaccia a riga di comando (CLI): la AWS CLI consente di interagire con i AWS servizi utilizzando una shell a riga di comando. Gli strumenti forniscono l'accesso diretto al pubblico ai AWS servizi e ai comandi personalizzati disponibili APIs per un servizio. Scarica la AWS CLI.
- AWS CloudFormationperAmazon GameLift Servers: Il AWS CloudFormation servizio ti aiuta a modellare e configurare AWS le risorse per semplificare l'implementazione e la gestione dell'infrastruttura. Crea un AWS CloudFormation modello per descrivere le Amazon GameLift Servers risorse per la tua soluzione di hosting, quindi utilizza il modello per creare risorse aggiuntive o aggiornare le configurazioni. Visualizza il riferimento al tipo di Amazon GameLift Servers risorsa.

Per Amazon GameLift ServersRealtime

Configura e distribuisci Realtime server per ospitare le tue partite multiplayer. Per consentire ai client di gioco di connettersi ai Realtime server, utilizza l'SDK del Amazon GameLift Servers Realtime client. Per iniziare, scarica l'SDK del Realtime client. Per informazioni sulla configurazione, consultaIntegrazione di un client di gioco per Amazon GameLift ServersRealtime.

Supporto SDK

Il client SDK Realtime contiene la sorgente per i seguenti linguaggi:

- C# (.NET)

Ambienti di sviluppo

Compila l'SDK dal codice sorgente in base alle esigenze per i seguenti sistemi operativi di sviluppo e motori di gioco supportati:

- Sistemi operativi: Windows, Linux, Android, iOS
- Motori di gioco: Unity, motori che supportano le librerie C#

Sistemi operativi dei server di gioco

Puoi distribuire Realtime server su risorse di hosting che funzionano sulle seguenti piattaforme:

- Amazon Linux
- Amazon Linux 2

> **Note**
>
> AL2 sta per terminare il supporto. Scopri maggiori dettagli in Amazon Linux 2 FAQs.

Tutorial: onboarding rapido con il wrapper Amazon GameLift Servers

Benvenuto nel tutorial di onboarding per. Amazon GameLift Servers In questo tutorial, implementerai rapidamente il tuo server di gioco in modo che sia ospitato su una flotta di risorse di calcolo basate sul cloud. Usa questo tutorial per evitare di integrare l'SDK del server Amazon GameLift Servers nel codice del gioco e distribuisci invece il gioco con le funzionalità minime necessarie per comunicare con il servizio ed eseguire sessioni di gioco. Amazon GameLift Servers Configurerai una soluzione di hosting di base e la utilizzerai per provare l'intera gamma di funzionalità come il ridimensionamento automatico e il matchmaking. È anche un ottimo modo per ospitare un prototipo del tuo gioco come parte di una demo dal vivo o per un test.

Principali vantaggi di questo metodo di onboarding:

- Implementa rapidamente il tuo server di gioco per un hosting rapido.
- Zero modifiche al codice di gioco e nessuna modifica richiesta.
- Usa questo metodo con qualsiasi file eseguibile di gioco, indipendentemente dal motore di gioco.

- Esplora tutti gli strumenti di Amazon GameLift Servers gestione, incluso il monitoraggio dell'attività delle sessioni di gioco e lo stato di salute dell'host.

> **Note**
>
> Il wrapper è destinato alla valutazione e all'uso di base nella produzione. Le funzionalità avanzate, come la gestione dettagliata delle sessioni dei giocatori, richiedono un'integrazione completa dell'SDK del server.

Prerequisiti

Prima di iniziare, assicurati di avere:

- E Account AWS con le autorizzazioni appropriate
- AWS CLI installato
- Vai a 1.18+
- Un server di gioco multiplayer eseguibile
- Crea (Linux/Mac)
- Git installato con un account attivo

Panoramica

In questo tutorial potrai:

1. Prendi e costruisci l'involucro
2. Prepara la build del gioco
3. Configura il wrapper
4. Carica la build del server di gioco
5. Crea la EC2 flotta gestita
6. Crea e connettiti a una sessione di gioco
7. Monitora e gestisci i tuoi server di gioco
8. Ridimensiona i tuoi server di gioco

Passaggio 1: Ottieni e crea il wrapper del server di gioco

Usa i seguenti comandi per ottenere il sorgente del wrapper del server di gioco e creare il wrapper. Questi comandi usano SSH, ma puoi anche accedere direttamente al repository Github.

Windows

```
> git clone git@github.com:amazon-gamelift/amazon-gamelift-servers-game-server-
wrapper.git
> cd amazon-gamelift-servers-game-server-wrapper
> powershell -file .\build.ps1
```

Mac e Linux

```
$ git clone git@github.com:amazon-gamelift/amazon-gamelift-servers-game-server-
wrapper.git
$ cd amazon-gamelift-servers-game-server-wrapper
$ make
```

In caso di compilazione riuscita, viene aggiunta una directory «out» a`amazon-gamelift-servers-game-server-wrapper`. In questa directory ci sono tre cartelle, una per ogni opzione della flotta di hosting supportata, che contengono un set di elementi di build. Per questo tutorial, eseguirai la distribuzione su EC2 flotte gestite, quindi utilizzerai la cartella. `gamelift-servers-managed-ec2`

Passaggio 2: prepara la build del server di gioco

In questo passaggio, prepari i file di build del server di gioco da caricare su. Amazon GameLift Servers

Crea la directory del gioco

Ora prepara una directory di gioco sul tuo computer locale. Questa directory deve contenere tutti i file necessari per far funzionare il server di giocoAmazon GameLift Servers. Ciò include il wrapper del server di gioco, la build del server di gioco e il file config.yaml che fa funzionare il wrapper con il server di gioco.

Utilizza le fasi seguenti:

1. EC2 Flotta gestita. Nella cartella di output del wrapper del server di gioco, trova l'artefatto di build da distribuire in una flotta gestita. EC2 La build verrà scritta in una directory esterna come indicato qui:. `out\linux\amd64\gamelift-servers-managed-ec2`
2. Copia il file eseguibile del server di gioco e tutti i file associati necessari per essere eseguiti nella `gamelift-servers-managed-ec2` cartella. Se necessario, puoi avere cartelle annidate.

Un esempio di struttura di cartelle sarà simile a questo:

```
gamelift-servers-managed-ec2
#-- config.yaml
#-- amazon-gamelift-servers-game-server-wrapper
#-- MyGame
#   #-- server-executable.exe
#   #-- my-game-settings
#   # ......
```

Fase 3: Configura il wrapper per la tua flotta

Amazon GameLift Serversgestisce il ciclo di vita delle istanze di calcolo di una flotta, creando nuove istanze con la build del server installata e riciclando le istanze secondo necessità. Il servizio gestisce il ciclo di vita dei processi del server di gioco che viene eseguito su ogni istanza. Una EC2 flotta gestita può avere istanze in più sedi per supportare i giocatori ovunque si trovino.

Modifica il file config.yaml per configurare il wrapper per la registrazione, la configurazione delle porte e l'inizializzazione del server.

1. Configura le impostazioni di registrazione. Il wrapper del server di gioco genera messaggi di registro per ogni processo del server di gioco. Per impostazione predefinita, il livello di registro è impostato su debug per la massima verbosità. Ciò è molto utile durante la configurazione e la risoluzione dei problemi e determina il livello di dettaglio dei messaggi di registro, in questo caso i più dettagliati. Le opzioni includono debug, info, warn ed error (meno dettagliato).
2. Specificate il percorso della directory di registro del server di gioco. Il percorso predefinito per i log del server di gioco è `./game-server-logs`. Questa directory contiene tutti i log generati dal server di gioco e ogni istanza li contiene. I log vengono caricati automaticamente nella posizione Amazon GameLift Servers in cui sono accessibili dalla scheda Eventi. Visualizza la sezione Risoluzione dei problemi per maggiori dettagli.

3. Definire la configurazione delle porte di rete. Imposta la porta di gioco come preferisci. Per questo tutorial, specifica una sola porta, poiché creerai una flotta che esegue solo un processo simultaneo del server di gioco per istanza. Se decidi di eseguire più processi contemporaneamente, dovrai configurare un numero sufficiente di porte per ogni processo simultaneo. Il valore predefinito è 37016, come mostrato nel file di configurazione, ma in generale, per le flotte che utilizzano build Linux, utilizza le porte 22 e 1026-60000. Per le flotte che utilizzano build Windows, utilizzate le porte 1026-60000.
4. Imposta il percorso dell'eseguibile del server di gioco. Per `./MyGame/my-server-executable.exe` personalizzare il percorso dell'eseguibile del server di gioco con il nome e la posizione effettivi. Questo è il punto di accesso per avviare il server di gioco.
5. Configura gli argomenti del server di gioco. Come minimo, specifica un argomento -port e usa lo stesso valore di porta di gioco che hai definito in precedenza. Il valore «pos» 0 indica che questo è il primo argomento. Aggiungi altri argomenti se necessario. Questi argomenti vengono passati al server di gioco all'avvio, consentendoti di configurarne il comportamento in fase di esecuzione.
   a. Argomento: `"--port"`
   b. Valore: `"{{.port number here}}"`
   c. Posizione: `0` (primo argomento nell'elenco)

Configurazione di esempio:

```
log-config:
  wrapper-log-level: debug
game-server-logs-dir: ./game-server-logs
ports:
  gamePort: 37016
game-server-details:
  executable-file-path: ./MyGame/my-server-executable
game-server-args:
  - arg: "--port"
    val: "{{.gameport}}"
    pos: 0
```

Passaggio 4: carica la build del server di gioco

Ora hai completato tutti gli elementi richiesti della build del tuo server di gioco (game server wrapper, config.yaml e i file del server di gioco) e sei pronto per caricare la build del gioco su come hosting. Amazon GameLift Servers Il modo più rapido per caricare la build del gioco è usare il AWS CLI comando, come mostrato nell'esempio seguente`upload-build`.

Caricare una build di gioco con Windows:

```
aws gamelift upload-build \
    --name gamelift-test-2025-03-11-1 \
    --build-version gamelift-test-2025-03-11-1 \
    --build-root out/windows/amd64/gamelift-servers-managed-ec2 \
    --operating-system WINDOWS_2016 \
    --server-sdk-version 5.3.0 \
    --region us-west-2
```

> **Note**
>
> Per le versioni Mac e Linux, usa --operating-system AMAZON_LINUX_2023

Quando crei la build, registra l'ID di build dalla risposta dell'API per utilizzarlo per la creazione della flotta.

Fase 5: Creare la EC2 flotta gestita

I passaggi seguenti descrivono una configurazione minima del parco veicoli, in modo da poter essere operativi il prima possibile.

Per creare la tua flotta:

1. Accedi a AWS Management Console e vai aAmazon GameLift Servers.
2. Nella barra dei menu nella parte superiore della finestra della console, verifica in quale regione si trova la build. Prendine nota perché la tua flotta deve trovarsi nella stessa regione o non potrai trovare o scegliere la tua build.
3. Nel pannello di navigazione della EC2 sezione Gestita, scegli Builds.

4. Seleziona la build che hai caricato in precedenza per visualizzare la pagina dei dettagli della build.
5. Nella sezione Flotte, scegli Crea flotta, che visualizza la pagina Definisci i dettagli della EC2 flotta gestita, dalla quale puoi monitorare lo stato della tua flotta e visualizzare gli eventi di creazione della flotta nella scheda Eventi.
6. Compila il nome e la descrizione e scegli Avanti.
7. Nella pagina Definisci i dettagli dell'istanza, l'area della build viene mostrata per impostazione predefinita. Scegli le aree aggiuntive che desideri aggiungere.
8. Per Tipo di flotta scegli On-Demand.
9. In Tipi di istanze scegli c5.large e scegli Avanti.
10. Nella configurazione Runtime, poiché la build del gioco caricata utilizza il wrapper, devi invece puntare all'eseguibile wrapper. Per i server di gioco Windows questo è. `C:\game\amazon-gamelift-servers-game-server-wrapper.exe` Per i server di gioco Linux, questo è`/local/game/amazon-gamelift-servers-game-server-wrapper`.

    Ad esempio: `LaunchPath": "/local/game/amazon-gamelift-servers-game-server-wrapper", "ConcurrentExecutions": 1, "Parameters": "–port 37016`

    Inoltre, configura i valori della porta di gioco con un intervallo che tenga conto della porta impostata nei `config.yaml` parametri di avvio della configurazione di runtime. `config.yaml`Non è necessario che la porta in ingresso corrisponda a quella specificata nella configurazione di runtime della flotta, ma in fase di esecuzione se la configurazione specifica una porta diversa, quel valore prevale su quello contenuto in. `config.yaml` I parametri di avvio immessi nella configurazione di runtime hanno inoltre la precedenza su ciò che è contenuto in. `config.yaml`
11. Nella pagina Rivedi e crea, ricontrolla tutte le configurazioni, quindi scegli Invia per creare la tua flotta. Lo stato della flotta cambierà man mano che aumenterà la capacità di ospitare il server di gioco e molto presto mostrerà lo stato Attivo. Una volta completata l'attivazione e schierata la flotta, il servizio avvia il wrapper, pronto a ricevere una richiesta di sessione di gioco.

Passaggio 6: Crea e connettiti a una sessione di gioco

Quando lo stato della tua flotta risulta Attivo, significa che i server di gioco sono pronti e in attesa di ospitare una sessione di gioco. Per iniziare una sessione di gioco, invia una richiesta di sessione di gioco al Amazon GameLift Servers servizio. Qui, utilizzerai il AWS CLI per effettuare questa richiesta.

> **Note**
>
> Tieni presente che creare una sessione di gioco utilizzando il AWS CLI è utile per testare e acquisire familiarità con il processo. Ad un certo punto, aggiungerai chiamate AWS SDK programmatiche al servizio di backend di gioco come parte del sistema di matchmaking o di posizionamento delle sessioni di gioco.

Usa quanto segue per creare una sessione di gioco:

```
aws gamelift create-game-session \
--fleet-id <FLEET_ID> \
--game-properties '[{"Key": "exampleProperty", "Value": "exampleValue"}]' \
--maximum-player-session-count 3 \
--region us-west-2
```

Puoi anche passare proprietà di gioco personalizzate all'eseguibile del tuo server. Vedi gli argomenti del server di gioco nel README per i dettagli. Quando riceve la create-game-session chiamata, Amazon GameLift Servers informa il wrapper di avviare l'eseguibile del server di gioco e avviare una sessione di gioco. Il contenuto `config.yaml` influisce sulla configurazione del server di gioco e i parametri di avvio impostati nella console determinano la configurazione della sessione di gioco stessa.

Formato di esempio per aggiungere le proprietà del gioco:

```
defaultArgs:
        - arg: "--port"
          val: "{{.GamePort}}"
          pos: 0
        - arg: "--ipAddress"
          val: "{{.IpAddress}}"
          pos: 1
        - arg: "--gameSessionId"
          val: "{{.GameSessionId}}"
          pos: 2
```

Fase 7: Gestisci e monitora la tua flotta

Ora che il tuo parco server di gioco è configurato e la sessione di gioco è iniziata, puoi gestirla e monitorarla dalla Amazon GameLift Servers console. Il modo migliore per farlo è accedere alla pagina dei dettagli della flotta, dove puoi modificare i dettagli della flotta o modificare la scalabilità e la capacità della flotta nella scheda Scaling. Consulta la sezione seguente sul ridimensionamento dei server di gioco.

Scegli la scheda Metriche per visualizzare i grafici che illustrano il monitoraggio delle metriche relative all'attività e all'hardware. Per i dettagli sui grafici delle metriche, scegli il link Informazioni accanto all'ID del tuo parco veicoli. Inoltre, puoi monitorare da vicino i tuoi server di gioco dalla scheda Metriche, ma puoi anche aggiungere allarmi a queste metriche nella dashboard. CloudWatch

Per accedere alla CloudWatch dashboard dalla console:

1. Digita CloudWatch "" nella barra di ricerca e selezionalo dall'elenco dei risultati della ricerca per visualizzare la CloudWatch panoramica.
2. Scorri verso il basso e scegli Visualizza GameLift dashboard per visualizzare i grafici basati sulle metriche chiave per le tue flotte e le sessioni di gioco.

Passaggio 8: ridimensiona i tuoi server di gioco

Per il passaggio successivo, configuri il ridimensionamento automatico. Con la scalabilità automatica puoi scalare dinamicamente la capacità della tua flotta in risposta all'attività dei server di gioco. Man mano che i giocatori arrivano e iniziano le sessioni di gioco, la scalabilità automatica aggiunge altre istanze e, man mano che la domanda dei giocatori diminuisce, la scalabilità automatica rilascia le istanze inutilizzate. Si tratta di un modo efficace per ridurre al minimo le risorse e i costi di hosting, garantendo al contempo un'esperienza di gioco fluida e veloce.

Mentre ti prepari per il lancio del gioco, ti consigliamo di impostare la scalabilità automatica per le tue flotte. La scalabilità automatica è consigliata come metodo efficace per ridurre al minimo le risorse e i costi di hosting, garantendo al contempo un'esperienza di gioco fluida e veloce.

Per impostare manualmente la capacità del parco veicoli

Per impostare manualmente la capacità del parco veicoli

1. Vai alla scheda Scaling nella pagina dei dettagli della tua flotta.
2. Seleziona una località e scegli Modifica.

3. Modifica il valore delle istanze desiderate e le impostazioni delle dimensioni minime e massime per scalarle oltre i valori correnti, quindi scegli Conferma.

> **Note**
>
> Utilizza l'impostazione del numero massimo di istanze come intervallo temporale per evitare costi e scalabilità eccessivi.

Per utilizzare il ridimensionamento automatico basato sulla destinazione

Per utilizzare il ridimensionamento automatico basato sulla destinazione

La scalabilità automatica basata sugli obiettivi (tracciamento degli obiettivi) collega il ridimensionamento della flotta alla percentuale di sessioni di gioco disponibili. Man mano che i giocatori si affollano a giocare al tuo gioco e le sessioni di gioco disponibili diminuiscono, il sistema risponde aggiungendo automaticamente altre istanze al parco istanze.

1. In Politica di scalabilità automatica basata su Target, scegli Aggiungi criterio e imposta la capacità della flotta in modo che cambi automaticamente quando raggiunge la soglia della percentuale di sessioni di gioco disponibili che hai impostato. Un buffer più ampio può gestire meglio le sovratensioni, far sì che i nuovi giocatori entrino rapidamente nei giochi, ma può anche comportare costi di hosting più elevati.
2. Scegli Conferma per accettare le modifiche.

La scalabilità automatica basata su regole offre un controllo più granulare, ad esempio la possibilità di collegare la scalabilità ad altre metriche del parco veicoli e di impostare soglie e risposte di scalabilità personalizzate. Offre opzioni potenti, ma richiede anche l'utilizzo della CLI e test significativi per capire come si comportano le regole personalizzate in azione. Questo tutorial si concentra sulla configurazione iniziale dell'approccio basato sugli obiettivi.

Risolvi i problemi più comuni

Di seguito sono riportati i problemi più comuni che potresti riscontrare durante l'esecuzione dei server e delle sessioni di gioco. Quando il server o la sessione di gioco non funzionano correttamente, la prima cosa da fare è controllare i log, che potrebbero rivelare uno dei problemi descritti di seguito per le nuove distribuzioni o i giochi in produzione.

Nei log viene spesso rivelato quanto segue:

- Il processo del server di gioco non può essere avviato. Potrebbe trattarsi di un errore nella configurazione del wrapper: verifica che il file abbia il percorso di avvio corretto e i parametri e gli argomenti di avvio corretti.
- La build del server di gioco non è eseguibile. È probabile che si tratti di un errore nel codice del gioco.
- I giocatori non possono connettersi alle sessioni di gioco. È probabile che si tratti di un errore di configurazione della porta.
- Connessioni lente o in ritardo. Rivedi le politiche e le soglie di scalabilità.
- Nessuna connessione. Verifica le regole e la configurazione dei porti per la tua flotta.

Per visualizzare i registri degli eventi della tua flotta Amazon GameLift Servers

Per visualizzare i registri degli eventi della tua flotta Amazon GameLift Servers

1. Aprire la console Amazon GameLift Servers.
2. Nella pagina Dettagli della flotta, scegli la scheda Eventi e scarica il registro. Puoi anche monitorare le attività e le metriche hardware relative allo stato del server di gioco e alle attivazioni delle sessioni di gioco dalla scheda Metriche.

Per visualizzare i registri delle sessioni di gioco

Per visualizzare i registri delle sessioni di gioco

1. Dalla console, apri la tua flotta e apri la scheda Sessioni di gioco.
2. Scegli un ID della sessione di gioco dall'elenco per visualizzare la relativa pagina Panoramica.
3. Scegli Scarica i registri per scaricare un file di registro localmente.

Per visualizzare i registri delle sessioni di gioco con la CLI, utilizza l'`GetGameSessionLogURL`API. Amazon GameLift Serversarchivia automaticamente i log per 14 giorni.

Puoi anche configurare CloudWatch i log di Amazon per la tua flotta. Ciò fornisce funzionalità di registrazione aggiuntive e integrazione con altri servizi di AWS monitoraggio.

Per l'accesso ai log in tempo reale o per periodi di conservazione prolungati tramite CloudWatch:

1. Nella parte superiore della dashboard della Amazon GameLift Servers console, digita "CloudWatch" nella barra di ricerca e selezionala dal menu a discesa dei risultati.
2. Vai ai gruppi di CloudWatch log e cerca sessioni specifiche. Il metodo più semplice è fare clic su Cerca tutto e filtrare utilizzando gameSessionId o ClientID.

Passaggi successivi

- Crea una flotta con più sedi per aggiungere hosting in altre sedi
- Aggiungi una coda per le sessioni di gioco per ottenere i migliori posizionamenti possibili per le sessioni di gioco in più sedi
- Sperimenta con il FlexMatch matchmaking creando un matchmaker e un set di regole per il tuo gioco
- Inizia a lavorare sulle funzionalità del client di gioco e dei componenti del servizio di backend, in modo che i giocatori possano effettuare richieste di iscrizione e connettersi direttamente alle sessioni di gioco
- Quando sei pronto, passa a una soluzione completamente integrata

Esplora con Amazon GameLift Servers plugin

Il Amazon GameLift Servers il plugin è un componente aggiuntivo completo per il tuo motore di gioco Unreal o Unity. Ti guida attraverso i passaggi di base per distribuire il gioco per l'hosting con Amazon GameLift Servers. Con il set di strumenti e i flussi di lavoro del plug-in, puoi lavorare nell'ambiente di sviluppo del tuo motore di gioco per preparare il server di gioco per l'hosting, configurare l'hosting su un computer locale per i test, creare un semplice servizio di backend e distribuire il tuo server di gioco su un hosting gestito basato su cloud.

Usa il plugin per provare a lavorare con Amazon GameLift Servers e ottieni una soluzione di hosting di giochi attiva e funzionante rapidamente. Puoi lavorare con risorse di gioco di esempio o con il tuo progetto di gioco. Il plugin automatizza una serie di passaggi in modo da poter creare rapidamente una soluzione di lavoro semplice. Una volta completati i flussi di lavoro guidati del plug-in, sarai in grado di connettere un client di gioco a sessioni di gioco ospitate dal vivo tramite Amazon GameLift Servers. Dopo aver utilizzato il plugin per creare una semplice soluzione di hosting, puoi personalizzare la soluzione per soddisfare le esigenze del tuo gioco.

Il plugin è disponibile per i seguenti motori di gioco:

- Unreal Engine
- Unità

Il plugin include questi componenti per ogni motore di gioco:

- Moduli plug-in per l'editor del motore di gioco. Una volta installato il plugin, un nuovo pulsante del menu principale consente di accedere a Amazon GameLift Servers funzionalità.
- Librerie per Amazon GameLift Servers API di servizio con funzionalità lato client.
- Librerie per Amazon GameLift Servers server SDK (versione 5).
- Asset di esempio da utilizzare per testare l'integrazione di un server.
- Configurazioni modificabili, sotto forma di AWS CloudFormation modelli, che definiscono la soluzione del server di gioco.

Argomenti

Workflow dei plugin

I passaggi seguenti descrivono un percorso tipico di preparazione e implementazione del progetto di gioco su Amazon GameLift Servers. Puoi completare questi passaggi utilizzando l'editor del motore di gioco e il codice del gioco.

1. Crea un profilo utente che si colleghi all'utente del tuo AWS account e fornisca le credenziali di accesso con le autorizzazioni da utilizzare Amazon GameLift Servers.
2. Imposta AWS le risorse correlate utilizzate dal plug-in nella soluzione di hosting (denominata «bootstrapping»).
3. Aggiungi il codice server al tuo progetto per stabilire la comunicazione tra un server di gioco in esecuzione e il Amazon GameLift Servers servizio.
4. Aggiungi un codice client al tuo progetto che consente ai client di gioco di inviare richieste a Amazon GameLift Servers per avviare nuove sessioni di gioco e poi connettersi ad esse.

5. Usa il flusso di lavoro Anywhere per configurare la tua workstation locale come computer Anywhere e ospitare il tuo server di gioco. Avvia il server e il client di gioco localmente tramite il plug-in, connettiti a una sessione di gioco e verifica l'integrazione.
6. Usa il EC2 flusso di lavoro gestito per caricare il tuo server di gioco su Amazon GameLift Servers e implementa una soluzione di cloud hosting semplice ma completa. Avvia il client di gioco localmente tramite il plug-in, richiedi una sessione di gioco, connettiti ad essa e gioca.

Quando lavori con il plugin, creerai e utilizzerai AWS risorse. Queste azioni potrebbero comportare addebiti sull' AWS account in uso. Se non l'hai mai fatto AWS, queste azioni potrebbero essere incluse nel piano AWS gratuito.

Amazon GameLift Serversplugin per Unreal Engine

Questo plugin aggiunge l'SDK e gli strumenti del server Amazon GameLift Servers C++ all'editor UE. Utilizza i flussi di lavoro guidati dell'interfaccia utente per integrare la funzionalità SDK del server nel tuo progetto di gioco e implementare una soluzione di Amazon GameLift Servers hosting per il tuo server di gioco.

Con il plug-in, puoi creare una soluzione di hosting funzionante di base e quindi ottimizzarla e personalizzarla secondo necessità. Configura una flotta Amazon GameLift Servers Anywhere con la tua workstation locale come host. Per l'hosting su cloud con flotte di container gestite EC2 o gestite, implementa il server di gioco con una soluzione completa per gestire le richieste di sessioni di gioco e le connessioni dei client.

Argomenti

- Installa il plugin per il tuo progetto di gioco Unreal
- Passaggi successivi: personalizza la tua soluzione di hosting di giochi
- Plugin per Unreal: configura un profilo AWS utente
- Plugin per Unreal: integra il tuo codice di gioco
- Plugin per Unreal: ospita il tuo gioco localmente con Anywhere Amazon GameLift Servers
- Plugin per Unreal: distribuisci il gioco su una flotta gestita EC2
- Plugin per Unreal: distribuisci il gioco su una flotta di container gestita

Installa il plugin per il tuo progetto di gioco Unreal

Scarica il Amazon GameLift Servers plugin per Unreal Engine da GitHub

Consulta il readme del GitHub repository per informazioni su come installare il plugin nel tuo Unreal Editor per un progetto di gioco.

Il plugin include questi componenti:

- Moduli plug-in per l'editor UE. Quando il plugin è installato, un nuovo pulsante del menu principale consente di accedere alle Amazon GameLift Servers funzionalità.
- Librerie C++ per l'API del Amazon GameLift Servers servizio. Utilizza la funzionalità API in un servizio di backend lato client per aiutare i client di gioco a richiedere sessioni di gioco e send/retrieve informazioni sulle sessioni di gioco.
- Librerie Unreal per l'SDK del Amazon GameLift Servers server (versione 5). Utilizza l'SDK del server nel codice del server di gioco per gestire la comunicazione tra i processi del server di gioco ospitato e il servizio. Amazon GameLift Servers
- Contenuti da testare, tra cui una mappa di gioco iniziale e due mappe di test con progetti di base ed elementi dell'interfaccia utente da utilizzare per testare l'integrazione di un server.
- Configurazioni modificabili, sotto forma di AWS CloudFormation modelli, utilizzate dal plug-in per distribuire il server di gioco per l'hosting.

Questo plugin utilizza AWS CloudFormation modelli per implementare soluzioni di hosting per scenari di gioco comuni. Utilizza queste soluzioni così come fornite o personalizzale secondo necessità per i tuoi giochi.

Passaggi successivi: personalizza la tua soluzione di hosting di giochi

L'utilizzo dei flussi di lavoro guidati del plug-in è un buon modo per iniziare a utilizzare rapidamente una soluzione di Amazon GameLift Servers hosting. Con il plug-in, puoi configurare le versioni di base di ciascuno dei componenti della tua soluzione.

Quando sei pronto, puoi sfruttare questa soluzione di base personalizzando ogni componente e perfezionandola mentre ti prepari al lancio del gioco. Prendi in considerazione queste opzioni:

- Modifica le flotte e le configurazioni della flotta. Consultare Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite.
- Personalizza la configurazione della coda delle sessioni di gioco. Vedi: Personalizza una coda di sessioni di gioco
- Aggiungi funzionalità al tuo server di gioco e al tuo client di gioco. Consulta Integra il tuo server di gioco con Amazon GameLift Servers e Integra il tuo client di gioco conAmazon GameLift Servers.

- Personalizza il tuo servizio di backend. Consultare Progetta il tuo servizio client di gioco.
- Imposta il ridimensionamento automatico della capacità per soddisfare la domanda prevista dei giocatori. Consultare Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers.
- Configura gli strumenti di osservabilità dell'hosting, tra cui analisi e registrazione. Consultare Monitoraggio Amazon GameLift Servers.
- Automatizza la distribuzione utilizzando Infrastructure as Code (IaC). I flussi di lavoro guidati del plug-in per le soluzioni gestite utilizzano modelli. AWS CloudFormation Puoi personalizzarli secondo necessità. Consultare Gestione di Amazon GameLift Servers hosting di risorse utilizzando AWS CloudFormation.

Argomenti

Plugin per Unreal: configura un profilo AWS utente

Dopo aver installato il plugin, configura un profilo utente con un AWS account valido. È possibile mantenere più profili nel plug-in, ma è possibile selezionare un solo profilo alla volta. Ogni volta che lavori nel plugin, seleziona un profilo da usare. Ogni pagina del flusso di lavoro mostra il profilo attualmente selezionato.

Il mantenimento di più profili consente di passare da una distribuzione di hosting all'altra. Ad esempio, è possibile configurare profili che utilizzano lo stesso AWS account ma vengono distribuiti in regioni diverse. AWS Oppure puoi configurare profili con AWS account o utenti e set di autorizzazioni diversi.

> **Note**
>
> Se hai installato la AWS CLI sulla tua workstation e hai già un profilo configurato, il Amazon GameLift Servers plugin lo rileverà e lo elencherà come profilo esistente. Il plugin seleziona automaticamente qualsiasi profilo denominato `[default]` È possibile utilizzare un profilo esistente o crearne uno nuovo.

Tutti i profili devono essere avviati per configurare alcune AWS risorse necessarie nell'account utente.

Per gestire i tuoi profili AWS

1. Nella barra degli strumenti principale dell'editor di Unreal, scegli il Amazon GameLift Servers menu e seleziona Credenziali di AWS accesso. Questa azione apre il Amazon GameLift Servers plugin alla pagina Configura il tuo profilo utente.
2. Utilizza i pulsanti per creare un nuovo AWS account o configurare un profilo utente per un AWS account che già possiedi.
3. Se non disponi già di un profilo utente, ti verrà richiesto di inserire i dettagli del profilo e creare un nuovo profilo. Inserisci le informazioni che seguono:

   - Un AWS account. Se crei un nuovo AWS account, usa il link AWS Management Console e segui le istruzioni. Vedi Creare un AWS account per maggiori dettagli.
   - Un AWS utente con autorizzazioni all'uso Amazon GameLift Servers e altri AWS servizi richiesti. Consulta Configura un Account AWS le istruzioni sulla configurazione di un utente AWS Identity and Access Management (IAM) con Amazon GameLift Servers autorizzazioni e accesso programmatico con credenziali a lungo termine.
   - Credenziali per il tuo utente. AWS Queste credenziali sono costituite da un ID della chiave di AWS accesso e da una chiave AWS segreta. Per maggiori dettagli, consulta Ottenere le chiavi di accesso.
   - AWS regione. Questa è una posizione geografica in cui desideri creare AWS le tue risorse per l'hosting. Durante lo sviluppo, ti consigliamo di utilizzare un'area vicina alla tua posizione fisica. Scegli una regione dall'elenco delle AWS regioni supportate.
4. Se il plugin rileva profili esistenti, visualizza un elenco di profili disponibili. Seleziona un profilo esistente dall'elenco o scegli Aggiungi un altro profilo per crearne uno nuovo.

Avvia un profilo utente

Tutti i profili devono essere avviati per essere utilizzati con il plugin. Amazon GameLift Servers Il bootstrap crea un bucket Amazon S3 specifico per il profilo. Viene utilizzato per archiviare configurazioni di progetto, creare artefatti e altre dipendenze. I bucket non vengono condivisi tra altri profili.

Il bootstrap implica la creazione di nuove AWS risorse e potrebbe comportare dei costi.

Per avviare il tuo profilo:

1. Nella pagina Credenziali di AWS accesso, controlla lo stato di bootstrap del profilo utente che desideri utilizzare. Se lo stato di bootstrap del profilo è «Inattivo» e non è presente alcun bucket S3 nell'elenco, devi avviare il profilo.
2. Seleziona il profilo che desideri utilizzare e scegli il profilo Bootstrap.
3. Attendi che lo stato del bootstrap passi a «Attivo». Ciò può richiedere alcuni minuti.

Plugin per Unreal: integra il tuo codice di gioco

Prima di poter distribuire il server di gioco su una flotta, devi apportare una serie di aggiornamenti al codice di gioco e impacchettare i componenti di gioco da utilizzare con il Amazon GameLift Servers servizio.

Questo argomento illustra i passaggi per eseguire un'integrazione minima. Per l'integrazione con il server, usa l'esempio di codice fornito per aggiornare la modalità di gioco del progetto.

- Imposta gli obiettivi di costruzione e le regole del modulo
- Aggiorna il codice del server di gioco
- Integra la mappa di gioco del tuo client
- Package dei componenti del gioco

Imposta gli obiettivi di costruzione e le regole del modulo

Modifica i file di progetto del gioco per generare correttamente i componenti di compilazione da utilizzare conAmazon GameLift Servers.

Per aggiungere obiettivi di build per client e server:

1. Apri i file di codice del progetto di gioco e individua il `.../Games/[your application name]Source/[your application name]Target.cs` file. Esempio: `.../Source/GameLiftUnrealAppTarget.cs`. (Se usi Visual Studio, apri il `.sln` file del progetto).
2. Copia questo file per creare due nuovi file di destinazione nella `Source/` directory.
    - Target client: rinomina il nuovo file in`[your application name]Client.Target.cs`. Modifica il contenuto per aggiornare i valori del nome della classe e del tipo di destinazione, come illustrato nel seguente codice di esempio:

```
using UnrealBuildTool;
  using System.Collections.Generic;

  public class GameLiftUnrealAppClientTarget :  TargetRules
 {
     public GameLiftUnrealAppClientTarget ( TargetInfo Target ) :  base
 ( Target )
     {
         Type = TargetType.Client;
         DefaultBuildSettings = BuildSettingsVersion.V2;
         IncludeOrderVersion = EngineIncludeOrderVersion.Unreal5_1;
         ExtraModuleNames.Add( "GameLiftUnrealApp");
     }
 }
```

- Destinazione del server: rinomina il nuovo file in. *[your application name]*`Server.Target.cs` Modifica il contenuto per aggiornare i valori del nome della classe e del tipo di destinazione, come illustrato nel seguente codice di esempio:

```
using UnrealBuildTool;
  using System.Collections.Generic;

  public class GameLiftUnrealAppServerTarget :  TargetRules
 {
     public GameLiftUnrealAppServerTarget ( TargetInfo Target ) :  base
 ( Target )
     {
         Type = TargetType.Server;
         DefaultBuildSettings = BuildSettingsVersion.V2;
         IncludeOrderVersion = EngineIncludeOrderVersion.Unreal5_1;
         ExtraModuleNames.Add( "GameLiftUnrealApp");
     }
 }
```

3. Rigenera i file di progetto. Se usi Visual Studio, puoi fare clic con il pulsante destro del mouse sul `.uproject` file del progetto di gioco e selezionare Genera file di progetto di Visual Studio.

Per aggiornare le regole del modulo del progetto di gioco:

Aggiorna le regole del modulo del progetto di gioco per fare in modo che dipenda dal plugin.

1. Apri i file di codice del progetto di gioco e individua il `.../Games/[your application name]Source/[your application name].Build.cs` file. Esempio: `.../Source/GameLiftUnrealApp.Build.cs`. (Se usi Visual Studio, apri il `.sln` file del progetto).
2. Individua la `ModuleRules` classe e aggiorna come illustrato nel seguente codice di esempio:

```
using UnrealBuildTool;

 public class GameLiftUnrealApp :  ModuleRules
{
    public GameLiftUnrealApp ( ReadOnlyTargetRules Target ) :  base ( Target )
    {
        PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;

        PublicDependencyModuleNames.AddRange( new string[] {  "Core",
"CoreUObject",  "Engine",  "InputCore",  "HeadMountedDisplay",  "EnhancedInput",
"GameLiftServerSDK" });

        bEnableExceptions =  true;
    }
}
```

3. Dopo aver creato i nuovi file di destinazione e modificato le regole del modulo, ricostruisci il progetto del gioco.

Aggiorna il codice del server di gioco

Aggiorna il codice del server di gioco per abilitare la comunicazione tra un processo del server di gioco e il Amazon GameLift Servers servizio. Il server di gioco deve essere in grado di rispondere alle richieste provenientiAmazon GameLift Servers, ad esempio, dall'avvio e dall'interruzione di nuove sessioni di gioco.

Per aggiungere il codice del server per Amazon GameLift Servers

1. Nel tuo editor di codice, apri il file solution (`.sln`) per il tuo progetto di gioco, che di solito si trova nella cartella principale del progetto. Ad esempio: `GameLiftUnrealApp.sln`.
2. Con la soluzione aperta, individua il file di intestazione della modalità gioco del progetto: `[project-name]GameMode.h` file. Ad esempio: `GameLiftUnrealAppGameMode.h`.
3. Modifica il file di intestazione per allinearlo al codice seguente. Assicurati di sostituire "GameLiftServer" con il nome del tuo progetto. Questi aggiornamenti sono specifici per il server

di gioco; ti consigliamo di creare una copia di backup dei file delle modalità di gioco originali da utilizzare con il tuo client.

Esempio di codice GameMode.h

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

#pragma once

#include "CoreMinimal.h"
#include "GameFramework/GameModeBase.h"
#include "GameLiftUnrealAppGameMode.generated.h"

struct FProcessParameters;

DECLARE_LOG_CATEGORY_EXTERN(GameServerLog, Log, All);

UCLASS(minimalapi)
class AGameLiftUnrealAppGameMode : public AGameModeBase
{
    GENERATED_BODY()

public:
    AGameLiftUnrealAppGameMode();

protected:
    virtual void BeginPlay() override;

private:
    void InitGameLift();

private:
    TSharedPtr<FProcessParameters> ProcessParameters;
};
```

- Apri il `[project-name]GameMode.cpp` file sorgente correlato (ad esempio). `GameLiftUnrealAppGameMode.cpp` Modificate il codice per allinearlo al codice di esempio seguente. Assicurati di sostituire "GameLiftUnrealApp" con il nome del tuo progetto. Questi aggiornamenti sono specifici per il server di gioco; ti consigliamo di creare una copia di backup del file originale da utilizzare con il tuo client.

Il codice di esempio seguente mostra come aggiungere gli elementi minimi richiesti per l'integrazione del server conAmazon GameLift Servers:

- Inizializza un client Amazon GameLift Servers API. La `InitSDK()` chiamata con i parametri del server è necessaria per una flotta Amazon GameLift Servers Anywhere. Quando ti connetti a una flotta Anywhere, il plug-in memorizza i parametri del server come argomenti della console. Il codice di esempio può accedere ai valori in fase di esecuzione.
- Implementa le funzioni di callback richieste per rispondere alle richieste del Amazon GameLift Servers servizio, tra cui `OnStartGameSession`OnProcessTerminate, e`onHealthCheck`.
- Chiama `ProcessReady()` con una porta designata per avvisare il Amazon GameLift Servers servizio quando è pronto per ospitare sessioni di gioco.

Esempio di codice del server di gioco

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

#include "GameLiftUnrealAppGameMode.h"

#include "UObject/ConstructorHelpers.h"
#include "Kismet/GameplayStatics.h"

#if WITH_GAMELIFT
#include "GameLiftServerSDK.h"
#include "GameLiftServerSDKModels.h"
#endif

#include "GenericPlatform/GenericPlatformOutputDevices.h"

DEFINE_LOG_CATEGORY(GameServerLog);

AGameLiftUnrealAppGameMode::AGameLiftUnrealAppGameMode() :
    ProcessParameters(nullptr)
{
    // Set default pawn class to our Blueprinted character
    static ConstructorHelpers::FClassFinder<APawn> PlayerPawnBPClass(TEXT("/Game/
ThirdPerson/Blueprints/BP_ThirdPersonCharacter"));

    if (PlayerPawnBPClass.Class != NULL)
    {
```

```
        DefaultPawnClass = PlayerPawnBPClass.Class;
    }

    UE_LOG(GameServerLog, Log, TEXT("Initializing AGameLiftUnrealAppGameMode..."));
}

void AGameLiftUnrealAppGameMode::BeginPlay()
{
    Super::BeginPlay();

#if WITH_GAMELIFT
    InitGameLift();
#endif
}

void AGameLiftUnrealAppGameMode::InitGameLift()
{
#if WITH_GAMELIFT
    UE_LOG(GameServerLog, Log, TEXT("Calling InitGameLift..."));

    // Getting the module first.
    FGameLiftServerSDKModule* GameLiftSdkModule =
 &FModuleManager::LoadModuleChecked<FGameLiftServerSDKModule>(FName("GameLiftServerSDK"));

    //Define the server parameters for a GameLift Anywhere fleet. These are not needed
 for a GameLift managed EC2 fleet.
    FServerParameters ServerParametersForAnywhere;

    bool bIsAnywhereActive = false;
    if (FParse::Param(FCommandLine::Get(), TEXT("glAnywhere")))
    {
        bIsAnywhereActive = true;
    }

    if (bIsAnywhereActive)
    {
        UE_LOG(GameServerLog, Log, TEXT("Configuring server parameters for
 Anywhere..."));

        // If GameLift Anywhere is enabled, parse command line arguments and pass them
 in the ServerParameters object.
        FString glAnywhereWebSocketUrl = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereWebSocketUrl="),
 glAnywhereWebSocketUrl))
```

```
        {
            ServerParametersForAnywhere.m_webSocketUrl =
TCHAR_TO_UTF8(*glAnywhereWebSocketUrl);
        }

        FString glAnywhereFleetId = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereFleetId="),
glAnywhereFleetId))
        {
            ServerParametersForAnywhere.m_fleetId = TCHAR_TO_UTF8(*glAnywhereFleetId);
        }

        FString glAnywhereProcessId = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereProcessId="),
glAnywhereProcessId))
        {
            ServerParametersForAnywhere.m_processId =
TCHAR_TO_UTF8(*glAnywhereProcessId);
        }
        else
        {
            // If no ProcessId is passed as a command line argument, generate a
randomized unique string.
            FString TimeString = FString::FromInt(std::time(nullptr));
            FString ProcessId = "ProcessId_" + TimeString;
            ServerParametersForAnywhere.m_processId = TCHAR_TO_UTF8(*ProcessId);
        }

        FString glAnywhereHostId = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereHostId="),
glAnywhereHostId))
        {
            ServerParametersForAnywhere.m_hostId = TCHAR_TO_UTF8(*glAnywhereHostId);
        }

        FString glAnywhereAuthToken = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereAuthToken="),
glAnywhereAuthToken))
        {
            ServerParametersForAnywhere.m_authToken =
TCHAR_TO_UTF8(*glAnywhereAuthToken);
        }

        FString glAnywhereAwsRegion = "";
```

```
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereAwsRegion="),
glAnywhereAwsRegion))
        {
            ServerParametersForAnywhere.m_awsRegion =
TCHAR_TO_UTF8(*glAnywhereAwsRegion);
        }

        FString glAnywhereAccessKey = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereAccessKey="),
glAnywhereAccessKey))
        {
            ServerParametersForAnywhere.m_accessKey =
TCHAR_TO_UTF8(*glAnywhereAccessKey);
        }

        FString glAnywhereSecretKey = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereSecretKey="),
glAnywhereSecretKey))
        {
            ServerParametersForAnywhere.m_secretKey =
TCHAR_TO_UTF8(*glAnywhereSecretKey);
        }

        FString glAnywhereSessionToken = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereSessionToken="),
glAnywhereSessionToken))
        {
            ServerParametersForAnywhere.m_sessionToken =
TCHAR_TO_UTF8(*glAnywhereSessionToken);
        }

        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_YELLOW);
        UE_LOG(GameServerLog, Log, TEXT(">>>> WebSocket URL: %s"),
*ServerParametersForAnywhere.m_webSocketUrl);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Fleet ID: %s"),
*ServerParametersForAnywhere.m_fleetId);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Process ID: %s"),
*ServerParametersForAnywhere.m_processId);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Host ID (Compute Name): %s"),
*ServerParametersForAnywhere.m_hostId);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Auth Token: %s"),
*ServerParametersForAnywhere.m_authToken);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Aws Region: %s"),
*ServerParametersForAnywhere.m_awsRegion);
```

```
        UE_LOG(GameServerLog, Log, TEXT(">>>> Access Key: %s"),
 *ServerParametersForAnywhere.m_accessKey);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Secret Key: %s"),
 *ServerParametersForAnywhere.m_secretKey);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Session Token: %s"),
 *ServerParametersForAnywhere.m_sessionToken);
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }

    UE_LOG(GameServerLog, Log, TEXT("Initializing the GameLift Server..."));

    //InitSDK will establish a local connection with GameLift's agent to enable further
 communication.
    FGameLiftGenericOutcome InitSdkOutcome = GameLiftSdkModule-
>InitSDK(ServerParametersForAnywhere);
    if (InitSdkOutcome.IsSuccess())
    {
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_GREEN);
        UE_LOG(GameServerLog, Log, TEXT("GameLift InitSDK succeeded!"));
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }
    else
    {
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_RED);
        UE_LOG(GameServerLog, Log, TEXT("ERROR: InitSDK failed : ("));
        FGameLiftError GameLiftError = InitSdkOutcome.GetError();
        UE_LOG(GameServerLog, Log, TEXT("ERROR: %s"), *GameLiftError.m_errorMessage);
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
        return;
    }

    ProcessParameters = MakeShared<FProcessParameters>();

    //When a game session is created, Amazon GameLift Servers sends an activation
 request to the game server and passes along the game session object containing game
 properties and other settings.
    //Here is where a game server should take action based on the game session object.
    //Once the game server is ready to receive incoming player connections, it should
 invoke GameLiftServerAPI.ActivateGameSession()
    ProcessParameters->OnStartGameSession.BindLambda([=]
(Aws::GameLift::Server::Model::GameSession InGameSession)
        {
            FString GameSessionId = FString(InGameSession.GetGameSessionId());
```

```
            UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"),
*GameSessionId);
            GameLiftSdkModule->ActivateGameSession();
        });

    //OnProcessTerminate callback. Amazon GameLift Servers will invoke this callback
before shutting down an instance hosting this game server.
    //It gives this game server a chance to save its state, communicate with services,
etc., before being shut down.
    //In this case, we simply tell Amazon GameLift Servers we are indeed going to
shutdown.
    ProcessParameters->OnTerminate.BindLambda([=]()
        {
            UE_LOG(GameServerLog, Log, TEXT("Game Server Process is terminating"));
            GameLiftSdkModule->ProcessEnding();
        });

    //This is the HealthCheck callback.
    //Amazon GameLift Servers will invoke this callback every 60 seconds or so.
    //Here, a game server might want to check the health of dependencies and such.
    //Simply return true if healthy, false otherwise.
    //The game server has 60 seconds to respond with its health status. Amazon GameLift
Servers will default to 'false' if the game server doesn't respond in time.
    //In this case, we're always healthy!
    ProcessParameters->OnHealthCheck.BindLambda([]()
        {
            UE_LOG(GameServerLog, Log, TEXT("Performing Health Check"));
            return true;
        });

    //GameServer.exe -port=7777 LOG=server.mylog
    ProcessParameters->port = FURL::UrlConfig.DefaultPort;
    TArray<FString> CommandLineTokens;
    TArray<FString> CommandLineSwitches;

    FCommandLine::Parse(FCommandLine::Get(), CommandLineTokens, CommandLineSwitches);

    for (FString SwitchStr : CommandLineSwitches)
    {
        FString Key;
        FString Value;

        if (SwitchStr.Split("=", &Key, &Value))
        {
```

```
            if (Key.Equals("port"))
            {
                ProcessParameters->port = FCString::Atoi(*Value);
            }
        }
    }

    //Here, the game server tells Amazon GameLift Servers where to find game session
 log files.
    //At the end of a game session, Amazon GameLift Servers uploads everything in the
 specified
    //location and stores it in the cloud for access later.
    TArray<FString> Logfiles;
    Logfiles.Add(TEXT("GameServerLog/Saved/Logs/GameServerLog.log"));
    ProcessParameters->logParameters = Logfiles;

    //The game server calls ProcessReady() to tell Amazon GameLift Servers it's ready
 to host game sessions.
    UE_LOG(GameServerLog, Log, TEXT("Calling Process Ready..."));
    FGameLiftGenericOutcome ProcessReadyOutcome = GameLiftSdkModule-
>ProcessReady(*ProcessParameters);

    if (ProcessReadyOutcome.IsSuccess())
    {
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_GREEN);
        UE_LOG(GameServerLog, Log, TEXT("Process Ready!"));
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }
    else
    {
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_RED);
        UE_LOG(GameServerLog, Log, TEXT("ERROR: Process Ready Failed!"));
        FGameLiftError ProcessReadyError = ProcessReadyOutcome.GetError();
        UE_LOG(GameServerLog, Log, TEXT("ERROR: %s"),
 *ProcessReadyError.m_errorMessage);
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }

    UE_LOG(GameServerLog, Log, TEXT("InitGameLift completed!"));
#endif
}
```

Integra la mappa di gioco del tuo client

La mappa di gioco di avvio contiene la logica del progetto e gli elementi dell'interfaccia utente che includono già il codice di base per richiedere sessioni di gioco e utilizzare le informazioni di connessione per connettersi a una sessione di gioco. Puoi usare la mappa così com'è o modificarla secondo necessità. Usa la mappa di gioco di avvio con altre risorse di gioco, come il modello di progetto Third Person fornito da Unreal Engine. Queste risorse sono disponibili in Content Browser. Puoi usarle per testare i flussi di lavoro di distribuzione del plugin o come guida per creare un servizio di backend personalizzato per il tuo gioco.

La mappa di avvio presenta le seguenti caratteristiche:

- Include la logica sia per una flotta Anywhere che per una EC2 flotta gestita. Quando gestisci il tuo client, puoi scegliere di connetterti a entrambe le flotte.
- Le funzionalità del client includono trovare una sessione di gioco (`SearchGameSessions()`), creare una nuova sessione di gioco (`CreateGameSession()`) e partecipare direttamente a una sessione di gioco.
- Ottiene un ID giocatore univoco dal pool di utenti Amazon Cognito del tuo progetto (fa parte di una soluzione distribuita Anywhere).

Per utilizzare la mappa di gioco di avvio

1. Nell'editor UE, apri la pagina Impostazioni del progetto, mappe e modalità ed espandi la sezione Mappe predefinite.
2. Per Editor Startup Map, seleziona StartupMap "" dall'elenco a discesa. Potrebbe essere necessario cercare il file, che si trova in`... > Unreal Projects/[project-name]/Plugins/Amazon GameLift Servers Plugin Content/Maps`.
3. Per Game Default Map, seleziona la stessa StartupMap "" dall'elenco a discesa.
4. Per Server Default Map, seleziona "ThirdPersonMap». Questa è una mappa predefinita inclusa nel tuo progetto di gioco. Questa mappa è progettata per due giocatori.
5. Apri il pannello dei dettagli per la mappa predefinita del server. Imposta GameMode Override su «Nessuno».
6. Espandi la sezione Modalità predefinite e imposta la modalità di gioco Global Default Server sulla modalità di gioco che hai aggiornato per l'integrazione del server.

Dopo aver apportato queste modifiche al progetto, sei pronto per creare i componenti del gioco.

Package dei componenti del gioco

Per impacchettare le build del server di gioco e del client di gioco

1. Apri il progetto di gioco in una versione originale dell'editor di Unreal Engine.
2. Usa l'editor per impacchettare le build del client di gioco e del server.
    a. Scegli un bersaglio. Vai a Piattaforme, Windows e seleziona una delle seguenti opzioni:
        - Server: `[your-application-name]Server`
        - Client: `[your-application-name]Client`
    b. Avvia la compilazione. Vai a Platform, Windows, Package Project.

Ogni processo di confezionamento genera un file eseguibile: `[your-application-name]Client.exe` o`[your-application-name]Server.exe`.

Nel plugin, imposta i percorsi degli eseguibili di compilazione del client e del server sulla tua workstation locale.

Plugin per Unreal: ospita il tuo gioco localmente con Anywhere Amazon GameLift Servers

Usa questo flusso di lavoro per configurare la tua workstation locale come host di server di gioco utilizzando una flotta Anywhere. Puoi utilizzarlo per testare l'integrazione del server di gioco prima di distribuirlo su una flotta gestita basata su cloud. Può anche essere utile per i test locali durante lo sviluppo iterativo di giochi.

Per avviare il flusso di lavoro Amazon GameLift Servers Anywhere:

- Nella barra degli strumenti principale dell'editor Unreal, scegli il Amazon GameLift Servers menu e seleziona Host with Anywhere. Questa azione apre la pagina del plugin Deploy Anywhere, che presenta un processo in sei fasi per integrare, creare e avviare i componenti del gioco.

Passaggio 1: imposta il tuo profilo

Scegli il profilo che desideri utilizzare quando segui questo flusso di lavoro. Il profilo selezionato influisce su tutte le fasi del flusso di lavoro. Tutte le risorse create sono associate all' AWS account del profilo e collocate nella AWS regione predefinita del profilo. Le autorizzazioni dell'utente del profilo determinano l'accesso alle AWS risorse e alle azioni.

Per impostare un profilo utente

1. Seleziona un profilo dall'elenco a discesa dei profili disponibili. Se non hai ancora un profilo o desideri crearne uno nuovo, vai al GameLift menu Amazon e scegli Imposta profili AWS utente.
2. Se lo stato di bootstrap non è «Attivo», scegli il profilo Bootstrap e attendi che lo stato passi a «Attivo».

Passaggio 2: configura il codice del gioco

In questo passaggio, prepara le build del server di gioco e del client di gioco su cui lavorareAmazon GameLift Servers. Se non hai ancora integrato il codice di gioco, consultaPlugin per Unreal: integra il tuo codice di gioco. Inserisci i percorsi degli eseguibili di gioco sulla tua workstation locale.

- Server di gioco: integra il tuo server di gioco con l'SDK del server Amazon GameLift Servers e crea un pacchetto per la build del tuo server di gioco. Per istruzioni, consultare Plugin per Unreal: integra il tuo codice di gioco. Il server di gioco deve essere integrato con l'SDK del server per stabilire una comunicazione con il Amazon GameLift Servers servizio e rispondere alle richieste di avvio di nuove sessioni di gioco e accettare connessioni con i client di gioco.
- Client di gioco: è necessario almeno un client di gioco in grado di connettersi al server di gioco utilizzando l'indirizzo IP e le informazioni sulla porta. Se non hai ancora configurato i componenti del client di giocoAmazon GameLift Servers, puoi utilizzare lo AWS CLI strumento per richiedere manualmente nuove sessioni di gioco, ottenere informazioni di connessione e utilizzare tali informazioni per connettere il client di gioco.

  A un certo punto, avrai bisogno di un servizio di backend per inviare nuove richieste di sessioni di gioco al Amazon GameLift Servers servizio e inoltrare le informazioni di connessione a un client di gioco. Puoi utilizzare le mappe di test incluse nel plugin per aggiungere Amazon GameLift Servers funzionalità client al tuo progetto di gioco. Per assistenza nella creazione di una soluzione personalizzata, consultaAggiungi Amazon GameLift Servers al tuo client di gioco.

Fase 3: Connettiti a una flotta Anywhere

In questa fase, si designa una flotta Anywhere da utilizzare. Una flotta Anywhere definisce una raccolta di risorse di elaborazione, che possono essere posizionate ovunque, per l'hosting di server di gioco.

- Se l' AWS account che stai utilizzando attualmente dispone di flotte Anywhere esistenti, apri il campo a discesa Fleet name e scegli una flotta. Questo menu a discesa mostra solo le flotte Anywhere nella AWS regione per il profilo utente attualmente attivo.
- Se non ci sono flotte esistenti o desideri crearne una nuova, scegli Crea una nuova flotta Anywhere e fornisci un nome per la flotta.

Dopo aver scelto una flotta Anywhere per il tuo progetto, Amazon GameLift Servers verifica che lo stato della flotta sia attivo e visualizza l'ID della flotta. Puoi tenere traccia dello stato di avanzamento di questa richiesta nel registro di output dell'editor Unreal.

Fase 4: Registra la tua postazione di lavoro

In questo passaggio, registri la tua workstation locale come risorsa di elaborazione nella nuova flotta Anywhere.

Per registrare la workstation come elaborazione Anywhere

1. Inserisci un nome di calcolo per il tuo computer locale. Se aggiungi più di un computer nel parco dati, i nomi devono essere univoci.
2. Fornisci un indirizzo IP per il tuo computer locale. L'impostazione predefinita di questo campo è l'indirizzo IP pubblico della macchina. Puoi anche usare localhost (127.0.0.1) purché utilizzi il client e il server di gioco sulla stessa macchina.
3. Scegli Register compute. Puoi tenere traccia dello stato di avanzamento di questa richiesta nel registro di output dell'editor Unreal.

In risposta a questa azione, Amazon GameLift Servers verifica che sia in grado di connettersi al computer e restituisce informazioni sul calcolo appena registrato. Crea inoltre gli argomenti della console necessari agli eseguibili di gioco per inizializzare la comunicazione con il servizio. Amazon GameLift Servers

Passaggio 5: Generazione del token di autenticazione

I processi del server di gioco in esecuzione sul computer Anywhere richiedono un token di autenticazione per effettuare chiamate al Amazon GameLift Servers servizio. Il plug-in genera e memorizza automaticamente un token di autenticazione per la flotta Anywhere ogni volta che avvii il server di gioco dal plug-in. Il valore del token di autenticazione viene memorizzato come argomento della riga di comando, che il codice del server può recuperare in fase di esecuzione.

Gli esempi di codice sopra riportati consentono inoltre di utilizzare AWS Signature Version 4 (SigV4) per le richieste API. SigV4 è il protocollo di AWS firma per aggiungere informazioni di autenticazione alle richieste API.

Non è necessario intraprendere alcuna azione in questo passaggio.

Passaggio 6: Avvia il gioco

A questo punto, hai completato tutte le attività necessarie per avviare e giocare alla partita multiplayer su una workstation locale utilizzandoAmazon GameLift Servers.

Per giocare al gioco ospitato

1. Avvia il tuo server di gioco. Il server di gioco ti avviserà Amazon GameLift Servers quando sarà pronto per ospitare le sessioni di gioco.
2. Avvia il client di gioco e usa la nuova funzionalità per iniziare una nuova sessione di gioco. Questa richiesta viene inviata Amazon GameLift Servers tramite il nuovo servizio di backend. In rispostaAmazon GameLift Servers, chiama il server di gioco, in esecuzione sul computer locale, per avviare una nuova sessione di gioco. Quando la sessione di gioco è pronta per accettare giocatori, Amazon GameLift Servers fornisce le informazioni di connessione per consentire al client di gioco di partecipare alla sessione di gioco.

Plugin per Unreal: distribuisci il gioco su una flotta gestita EC2

In questo flusso di lavoro, distribuisci il gioco per l'hosting su risorse di elaborazione basate su cloud gestite da. Amazon GameLift Servers Carica la build del tuo server di gioco integrato nel servizio per la Amazon GameLift Servers distribuzione. Se non hai ancora integrato il codice di gioco, consultaPlugin per Unreal: integra il tuo codice di gioco. Una volta completato questo flusso di lavoro, avrai un client di gioco funzionante in grado di connettersi ai tuoi server di gioco nel cloud.

Per avviare il EC2 flusso di lavoro Amazon GameLift Servers gestito di Amazon:

- Nella barra degli strumenti principale dell'editor Unreal, scegli il Amazon GameLift Servers menu e seleziona Host with Managed. EC2 Questa azione apre la pagina del plug-in Deploy Amazon EC2 Fleet, che presenta un processo in sei fasi per integrare, creare, distribuire e avviare i componenti del gioco.

Passaggio 1: imposta il tuo profilo

Scegli il profilo che desideri utilizzare quando segui questo flusso di lavoro. Il profilo selezionato influisce su tutte le fasi del flusso di lavoro. Tutte le risorse create sono associate all' AWS account del profilo e collocate nella AWS regione predefinita del profilo. Le autorizzazioni dell'utente del profilo determinano l'accesso alle AWS risorse e alle azioni.

Per impostare un profilo utente

1. Seleziona un profilo dall'elenco a discesa dei profili disponibili. Se non hai ancora un profilo o desideri crearne uno nuovo, vai al GameLift menu Amazon e scegli Imposta profili AWS utente.
2. Se lo stato di bootstrap non è «Attivo», scegli il profilo Bootstrap e attendi che lo stato passi a «Attivo».

Passaggio 2: configura il codice del gioco

In questo passaggio, prepara le build del server di gioco e del client di gioco per utilizzarle con l'SDK del server Amazon GameLift Servers C++ per Unreal. Se non hai ancora integrato il codice di gioco e non hai ancora creato gli eseguibili per client e server di gioco, consulta. Plugin per Unreal: integra il tuo codice di gioco Inserisci i percorsi degli eseguibili di gioco sulla tua workstation locale.

In questa fase del flusso di lavoro, il plugin fornisce collegamenti alle istruzioni e al codice sorgente per configurare una versione sorgente di Unreal Editor. È necessario utilizzare la versione generata dal codice sorgente quando si creano i componenti client e server.

Dopo aver creato un server di gioco integrato con l'SDK del server, completa le seguenti attività per prepararlo al caricamento per l'hosting. Amazon GameLift Servers

Per preparare la build del server per la distribuzione su cloud (Windows)

Nella `WindowsServer` cartella, in cui l'editor Unreal archivia i file di build del server per impostazione predefinita, apporta le seguenti aggiunte:

1. Copia lo script di installazione della build del server nella radice della **`WindowsServer`** cartella. Lo script di installazione è incluso nel download del plugin. Cerca il file`[project-name]/Plugins/Resources/CloudFormation/extra_server_resources/install.bat`. Amazon GameLift Serversutilizza questo file per installare la build del server sui tuoi computer di hosting.

2. Copia il **`VC_redist.x64.exe`** file nella radice della **`WindowsServer`** cartella. Questo file è incluso nell'installazione di Visual Studio. Di solito si trova in`C:/Program Files (x86)/Microsoft Visual Studio/2019/Professional/VC/Redist/MSVC/v142`.
3. Aggiungi i file della libreria OpenSSL alla build del tuo server di gioco.

   > **Note**
   >
   > Se il tuo server di gioco è integrato con la versione 5.3.x o successiva dell'SDK del server, puoi saltare questo passaggio.

   Per i server di gioco integrati con la versione 5.2 o precedente dell'SDK del server, devi individuare e copiare manualmente le librerie. È necessario utilizzare la stessa versione di OpenSSL utilizzata dalla versione di Unreal Engine 5. Le build di gioco distribuite con le librerie OpenSSL sbagliate non saranno in grado di comunicare con il servizio. Amazon GameLift Servers

   a. Cerca le librerie OpenSSL nella fonte del tuo motore di gioco. La posizione varia a seconda dell'ambiente di sviluppo:

      In Windows:

      - `[ENGINE_ROOT_DIR]\Engine\Extras\ThirdPartyNotUE\libimobiledevice\x64\libssl-1_1-x64.dll`
      - `[ENGINE_ROOT_DIR]\Engine\Extras\ThirdPartyNotUE\libimobiledevice\x64\libcrypto-1_1-x64.dll`

      In Linux:

      - `Engine/Source/Thirdparty/OpenSSL/1.1.1`*`n`*`/include/libssl.so.1.1`
      - `Engine/Source/Thirdparty/OpenSSL/1.1.1`*`n`*`/include/libcrypto.so.1.1`

   b. Copia le librerie OpenSSL nella directory del pacchetto di build del gioco all'indirizzo. `<YourGame>/Binaries/Win64`

Per preparare la build del server per l'implementazione su cloud (Linux)

Per istruzioni più dettagliate sulla preparazione di un server di gioco creato per Linux, consulta Creazione dell'SDK del server Amazon GameLift Servers per Unreal Engine 5 su Amazon Linux.

1. Designate una directory di lavoro per organizzare i file di build. La struttura della directory di lavoro viene distribuita così com'è su ogni computer di hosting. Aggiungi il tuo server di gioco basato su Linux e tutti i file dipendenti.
2. Crea uno script di installazione della build del server nella radice della tua directory di lavoro. Se necessario, crea un `install.sh` file e aggiungi tutti i comandi necessari per installare correttamente la build del server di gioco. Amazon GameLift Serversutilizza questo file per installare la build del server su ogni risorsa EC2 di hosting.
3. Aggiungi i file della libreria OpenSSL alla build del tuo server di gioco.

   > **Note**
   >
   > Se il tuo server di gioco è integrato con la versione 5.3.x o successiva dell'SDK del server, puoi saltare questo passaggio.

   Per i server di gioco integrati con la versione 5.2 o precedente dell'SDK del server, devi individuare e copiare manualmente le librerie. È necessario utilizzare la stessa versione di OpenSSL utilizzata dalla versione di Unreal Engine 5. Le build di gioco distribuite con le librerie OpenSSL sbagliate non saranno in grado di comunicare con il servizio. Amazon GameLift Servers

   a. Cerca le librerie OpenSSL nella fonte del tuo motore di gioco. La posizione varia a seconda dell'ambiente di sviluppo:

      In Windows:

      - `[ENGINE_ROOT_DIR]\Engine\Extras\ThirdPartyNotUE\libimobiledevice\x64\libssl-1_1-x64.dll`
      - `[ENGINE_ROOT_DIR]\Engine\Extras\ThirdPartyNotUE\libimobiledevice\x64\libcrypto-1_1-x64.dll`

      In Linux:

- `Engine/Source/Thirdparty/OpenSSL/1.1.1n/include/libssl.so.1.1`
- `Engine/Source/Thirdparty/OpenSSL/1.1.1n/include/libcrypto.so.1.1`

b. Quando trovi le librerie OpenSSL, copiale nella directory del pacchetto di build del gioco all'indirizzo. `<YourGame>/Binaries/Linux`

Fase 3: Seleziona lo scenario di implementazione

In questo passaggio, scegli la soluzione di hosting del gioco che desideri implementare in questo momento. Puoi avere più distribuzioni del gioco, utilizzando uno qualsiasi degli scenari.

- Flotta a regione singola: distribuisce il server di gioco su un'unica flotta di risorse di hosting nella regione predefinita del profilo attivo. AWS Questo scenario è un buon punto di partenza per testare l'integrazione del server AWS e la configurazione della build del server. Implementa le seguenti risorse:
  - AWS fleet (On-Demand) con la build del server di gioco installata e funzionante.
  - Pool di utenti e client Amazon Cognito per consentire ai giocatori di autenticarsi e iniziare una partita.
  - Autorizzatore di gateway API che collega il pool di utenti con. APIs
  - Web ACl per limitare le chiamate eccessive dei giocatori al gateway API.
  - API gateway + funzione Lambda per consentire ai giocatori di richiedere uno slot di gioco. Questa funzione chiama `CreateGameSession()` se non ce ne sono disponibili.
  - API gateway + funzione Lambda per consentire ai giocatori di ottenere informazioni di connessione per la loro richiesta di gioco.
- FlexMatch flotta: distribuisce il tuo server di gioco su una serie di flotte e configura un FlexMatch matchmaker con le regole per creare partite tra giocatori. Questo scenario utilizza un hosting Spot a basso costo con una struttura multi-flotta e più sedi per una disponibilità duratura. Questo approccio è utile quando sei pronto per iniziare a progettare un componente matchmaker per la tua soluzione di hosting. In questo scenario, creerai le risorse di base per questa soluzione, che potrai personalizzare in seguito, se necessario. Implementa le seguenti risorse:
  - FlexMatch configurazione del matchmaking e regole di matchmaking impostate per accettare le richieste dei giocatori e formare partite.
  - Tre AWS flotte con la configurazione del server di gioco installata e funzionante in più località. Include due flotte Spot e una flotta On-Demand come backup.

- AWS coda per il posizionamento delle sessioni di gioco che soddisfa le richieste di partite proposte trovando la migliore risorsa di hosting possibile (in base a fattibilità, costo, latenza dei giocatori, ecc.) e avviando una sessione di gioco.
- Pool di utenti e client Amazon Cognito per consentire ai giocatori di autenticarsi e iniziare una partita.
- Autorizzatore di gateway API che collega il pool di utenti con. APIs
- Web ACl per limitare le chiamate eccessive dei giocatori al gateway API.
- API gateway + funzione Lambda per consentire ai giocatori di richiedere uno slot di gioco. Questa funzione chiama`StartMatchmaking()`.
- API gateway + funzione Lambda per consentire ai giocatori di ottenere informazioni di connessione per la loro richiesta di gioco.
- Tabelle Amazon DynamoDB per archiviare i ticket di matchmaking per i giocatori e le informazioni sulle sessioni di gioco.
- Argomento SNS + funzione Lambda per GameSessionQueue gestire gli eventi.

Passaggio 4: imposta i parametri di gioco

In questo passaggio, descrivi il gioco su cui caricarlo AWS;

- Nome della build del server: fornisci un nome significativo per la build del server di gioco. AWS usa questo nome per fare riferimento alla copia della build del server che viene caricata e utilizzata per le distribuzioni.
- Sistema operativo basato su server: inserisci il sistema operativo su cui è costruito il server. Indica il AWS tipo di risorse di calcolo da utilizzare per ospitare il gioco.
- Cartella del server di gioco: identifica il percorso della cartella di build del server locale.
- Build del server di gioco: identifica il percorso dell'eseguibile del server di gioco.
- Percorso del client di gioco: identifica il percorso dell'eseguibile del client di gioco.
- Output della configurazione del client: questo campo deve puntare a una cartella nella build del client che contiene la AWS configurazione. Cercalo nella seguente posizione:`[client-build]/[project-name]/Content/CloudFormation`.

Fase 5: Implementazione dello scenario

In questo passaggio, distribuisci il gioco su una soluzione di hosting cloud in base allo scenario di distribuzione scelto. Questo processo può richiedere diversi minuti per AWS la convalida della build del server, il provisioning delle risorse di hosting, l'installazione del server di gioco, l'avvio dei processi del server e la preparazione per ospitare sessioni di gioco.

Per iniziare la distribuzione, scegli Deploy. CloudFormation Puoi monitorare lo stato del tuo hosting di giochi qui. Per informazioni più dettagliate, puoi accedere alla console di AWS gestione AWS e visualizzare le notifiche degli eventi. Assicurati di accedere utilizzando lo stesso account, utente e AWS regione del profilo utente attivo nel plug-in.

Una volta completata la distribuzione, il server di gioco è installato su un' AWS EC2 istanza. Almeno un processo del server è in esecuzione ed è pronto per iniziare una sessione di gioco.

Passaggio 6: Avvia il client

A questo punto, hai completato tutte le attività necessarie per avviare e giocare al gioco multiplayer ospitato conAmazon GameLift Servers. Per giocare, avvia un'istanza del tuo client di gioco.

Se hai implementato lo scenario a flotta singola, puoi aprire una singola istanza client con un giocatore, accedere alla mappa del server e spostarti. Apri istanze aggiuntive del client di gioco per aggiungere un secondo giocatore alla stessa mappa di gioco del server.

Se hai implementato FlexMatch lo scenario, la soluzione attende che almeno due client vengano messi in coda per il posizionamento della sessione di gioco prima che i giocatori possano accedere alla mappa del server.

Plugin per Unreal: distribuisci il gioco su una flotta di container gestita

Usa questo flusso di lavoro guidato dai plug-in per creare un'immagine del contenitore per il tuo server di gioco e distribuirla su una soluzione di hosting basata su container. Se non hai ancora integrato il codice di gioco, consulta. Plugin per Unreal: integra il tuo codice di gioco Una volta completato con successo questo flusso di lavoro, il server di gioco containerizzato è in esecuzione nel cloud e puoi utilizzare il plug-in per avviare un client di gioco, connetterti a una sessione di gioco e giocare.

Prima di iniziare

Questo flusso di lavoro presuppone che tu abbia completato le seguenti attività.

- Integra il codice del server di gioco con l'SDK Amazon GameLift Servers del server. Il server di gioco ospitato deve essere in grado di comunicare con il Amazon GameLift Servers servizio in modo da poter rispondere alle richieste di avvio di nuove sessioni di gioco e segnalare lo stato della sessione di gioco. Se non hai completato questa operazione, ti consigliamo di seguire prima il flusso di lavoro del plug-in Host with Anywhere. Per istruzioni sulla preparazione del codice del server di gioco, consultaAggiorna il codice del server di gioco. Per una flotta di container gestita, devi integrare il gioco con la versione 5.2 o successiva dell'SDK del server.

  > **Note**
  >
  > Se utilizzi la mappa di gioco di avvio, questa operazione è già stata eseguita per te.

- Package del tuo file eseguibile del server di gioco per eseguirlo su Linux. Se stai sviluppando su Windows e stai integrando C++ Server SDK versione 5.2.x o precedente, dovrai utilizzare l'Unreal cross compile toolkit. In alternativa, puoi configurare uno spazio di lavoro Linux separato o utilizzare uno strumento come il sottosistema Windows per Linux (WSL).
- Raccogli i file da distribuire con la build del tuo server di gioco. Sul computer locale, crea una directory di lavoro per organizzare i file, che verrà incorporata nell'immagine del contenitore del server di gioco. Questi potrebbero includere dipendenze di gioco, uno script per avviare i server di gioco e altri processi all'avvio di un contenitore, ecc.
- Aggiungi i file della libreria OpenSSL per la build del tuo server di gioco.

  - > **Note**
    >
    > Se il tuo server di gioco è integrato con la versione 5.3.x o successiva dell'SDK del server, puoi saltare questo passaggio.

    Per i server di gioco integrati con la versione 5.2 o precedente dell'SDK del server, devi individuare e copiare manualmente le librerie. È necessario utilizzare la stessa versione di OpenSSL utilizzata dalla versione di Unreal Engine 5. Le build di gioco distribuite con le librerie OpenSSL sbagliate non saranno in grado di comunicare con il servizio. Amazon GameLift Servers

    a. Cerca le librerie OpenSSL nella fonte del tuo motore di gioco. La posizione varia a seconda dell'ambiente di sviluppo:

       In Windows:

- `[ENGINE_ROOT_DIR]\Engine\Extras\ThirdPartyNotUE\libimobiledevice\x64\libssl-1_1-x64.dll`
- `[ENGINE_ROOT_DIR]\Engine\Extras\ThirdPartyNotUE\libimobiledevice\x64\libcrypto-1_1-x64.dll`

In Linux:

- `Engine/Source/Thirdparty/OpenSSL/1.1.1n/include/libssl.so.1.1`
- `Engine/Source/Thirdparty/OpenSSL/1.1.1n/include/libcrypto.so.1.1`

b. Copia le librerie OpenSSL nella directory del pacchetto di build del gioco all'indirizzo. `<YourGame>/Binaries/Win64`

- Integra il codice del client di gioco con. Amazon GameLift Servers Un modo per completare questa operazione consiste nell'aggiungere una risorsa di esempio (inclusa nel plug-in) già integrata. Per indicazioni sulla preparazione del codice client di gioco, consultaIntegra la mappa di gioco del tuo client .
- Installa Docker sul tuo computer locale. È necessario che questo strumento sia installato se desideri che il plug-in crei immagini dei contenitori per te e le invii a un repository ECR. In alternativa, puoi eseguire queste attività manualmente e indicare al plug-in di utilizzare un'immagine del contenitore esistente. Per ulteriori informazioni sulla creazione manuale dell'immagine, consulta Creare un'immagine del contenitore per Amazon GameLift Servers.

Per avviare il flusso di lavoro dei contenitori Amazon GameLift Servers gestiti:

- Nella barra degli strumenti principale dell'editor Unreal, scegli il Amazon GameLift Servers menu e seleziona Contenitori gestiti. Questa azione apre la pagina del plug-in Host with Managed Containers, che presenta un step-by-step processo per creare un'immagine del contenitore con la build del server di gioco, distribuirla in una flotta di container e avviare il gioco.

Passaggio 0: imposta il tuo profilo

Questa sezione mostra il profilo utente attualmente selezionato. Verifica che il profilo utente corrente sia quello che desideri utilizzare per questo flusso di lavoro. Tutte le risorse create in questo flusso di lavoro sono associate all' AWS account del profilo e collocate nella AWS regione predefinita del profilo. Le autorizzazioni dell'utente del profilo determinano l'accesso alle AWS risorse e alle azioni.

Potrebbe essere necessario modificare il profilo utente selezionato se:

- Al momento non è selezionato alcun profilo.
- Vuoi selezionare un profilo diverso o creare un nuovo profilo.
- È necessario eseguire il bootstrap del profilo selezionato (se lo stato di bootstrap è inattivo).

Per impostare o modificare il profilo utente selezionato

- Nel Amazon GameLift Servers menu, scegli Open AWS Access Credentials.

Fase 1: Valutare la disponibilità del contenitore

Prima di distribuire il tuo server di gioco su una flotta di container, devi impacchettarlo in un'immagine del contenitore e archiviarlo in un repository Amazon ECR. Il plug-in può completare queste attività per te oppure puoi eseguirle manualmente. In questo passaggio, fornisci informazioni sullo stato dell'immagine del contenitore e dell'archivio ECR.

Usa le domande di valutazione per indicare al plugin quali passi deve compiere:

- Crea una nuova immagine del contenitore. Se scegli questa opzione, nel passaggio successivo ti verrà richiesta la posizione della directory di build del server di gioco e l'eseguibile della build. Il plugin utilizza un modello Dockerfile (fornito daAmazon GameLift Servers) e lo configura automaticamente per il gioco. È possibile visualizzare il modello all'indirizzo. Crea un'immagine del contenitore per Amazon GameLift Servers Dopo aver scelto questa opzione, indica dove desideri che il plugin memorizzi la nuova immagine:
  - Crea un nuovo repository Amazon ECR e inviaci l'immagine del contenitore. Il plug-in crea un repository ECR privato utilizzando l' AWS account e lo utilizza come impostazione predefinita Regione AWS nel profilo utente selezionato.
  - Invia l'immagine del contenitore a un repository Amazon ECR creato in precedenza. Se scegli questa opzione, il passaggio successivo ti chiederà di selezionare un repository Amazon ECR esistente da un elenco. L'elenco include tutti i repository Amazon ECR per l' AWS account e quelli predefiniti Regione AWS nel profilo utente selezionato. Puoi selezionare un repository pubblico o privato.
- Usa un'immagine del contenitore esistente. Se hai creato un'immagine manualmente, ti consigliamo di utilizzare il modello Dockerfile fornito daAmazon GameLift Servers, disponibile all'indirizzo. Crea un'immagine del contenitore per Amazon GameLift Servers Dopo aver scelto questa opzione, indica dove si trova l'immagine.

- Un'immagine generata da Docker archiviata localmente. Se scegli questa opzione, il plug-in crea un nuovo repository privato Amazon ECR e vi invia il file di immagine locale. Il passaggio successivo richiederà un ID immagine, che il plug-in utilizza per localizzare il file di immagine.
- Un'immagine del contenitore già archiviata in un repository Amazon ECR. Se scegli questa opzione, il passaggio successivo ti chiederà di selezionare un repository e un'immagine Amazon ECR esistenti da un elenco. L'elenco include tutti i repository Amazon ECR per l' AWS account e quelli predefiniti Regione AWS nel profilo utente selezionato. Puoi selezionare un repository pubblico o privato.

Fase 2: Configurare la distribuzione delle immagini

In questo passaggio, fornisci le informazioni necessarie al plug-in per distribuire l'immagine del container in una flotta di container. Questo passaggio richiede le seguenti informazioni:

- La posizione della build del server di gioco, dell'immagine del contenitore o del repository Amazon ECR, in base alle selezioni effettuate nella Fase 1.
- Lo scenario da utilizzare per la distribuzione dei container gestiti.
- Il percorso di output della configurazione del client. Seleziona la cartella nella build del client che contiene la tua AWS configurazione. Cercalo nella seguente posizione:`[client-build]/[project-name]/Content/CloudFormation`.
- Impostazioni di distribuzione opzionali. Questa sezione contiene le impostazioni di configurazione che il plugin utilizza per impostazione predefinita. È possibile modificarli o mantenere i valori predefiniti
  - Per impostazione predefinita, il nome del gioco è impostato sul nome del progetto di gioco. Tutte le AWS risorse create dal plugin fanno riferimento al valore del nome del gioco.
  - L'intervallo di porte, il limite di memoria e il limite di vCPU sono impostazioni di configurazione per la flotta di container. Per ulteriori informazioni sulla personalizzazione di questi valori, consulta la sezione Configurare le connessioni di rete dedicata all'intervallo delle porte di connessione e Imposta i limiti delle risorse ai limiti delle risorse.
  - Il tag dell'immagine del contenitore viene utilizzato per classificare le immagini del contenitore in Amazon ECR. Il valore predefinito è `unreal-gamelift-plugin`.
  - Nome del repository Amazon ECR. Puoi modificare questo campo per suggerire un nome personalizzato solo quando il plug-in sta creando un repository ECR per te. Il valore predefinito è `unreal-game lift-plugin-ecr-repository`.

Opzioni dello scenario di implementazione

Flotta di container in un'unica regione

Questo scenario distribuisce il tuo server di gioco su una singola flotta di container. È un buon punto di partenza per testare l'integrazione del server AWS e la configurazione del container. Implementa le seguenti risorse.

- Amazon GameLift ServersLa definizione del gruppo di container descrive come distribuire ed eseguire le immagini dei container su una flotta di container.
- Amazon GameLift Serversflotta di container (On-Demand) con il container del server di gioco installato e funzionante, con alias.
- Pool di utenti e client Amazon Cognito per consentire ai giocatori di autenticarsi e iniziare una partita.
- Autorizzatore API Gateway che collega il pool di utenti con APIs.
- Elenco di controllo degli accessi Web (ACL) per limitare le chiamate eccessive dei giocatori all'API Gateway.
- Servizio di backend per effettuare richieste al Amazon GameLift Servers servizio per conto dei client di gioco, ad esempio per richiedere sessioni di gioco e partecipare ai giochi:
  - API Gateway + funzione Lambda per consentire ai giocatori di richiedere uno slot per la sessione di gioco. Questa funzione chiama `CreateGameSession()` se non sono disponibili slot aperti.
  - API Gateway + funzione Lambda per consentire ai giocatori di ottenere informazioni di connessione per la loro richiesta di gioco.

Flotta di container a regione singola con FlexMatch

Questo scenario distribuisce il server di gioco su una flotta di container, configura il posizionamento delle sessioni di gioco, e imposta il matchmaking. FlexMatch Questo scenario è utile quando sei pronto per iniziare a progettare un matchmaker personalizzato per la tua soluzione di hosting. Utilizza questo scenario per creare le risorse di base per questa soluzione, che potrai personalizzare in seguito, se necessario. Implementa le seguenti risorse:

- Amazon GameLift Serversdefinizione del gruppo di container che descrive come distribuire ed eseguire le immagini dei container su una flotta di container.
- Amazon GameLift Serversflotta di container (On-Demand) con il container del server di gioco installato e funzionante, con alias.

- FlexMatchconfigurazione del matchmaking e regole di matchmaking impostate per accettare le richieste dei giocatori e formare partite.
- Amazon GameLift Serverscoda di sessione di gioco che soddisfa le richieste di partite proposte trovando la migliore risorsa di hosting possibile (in base a fattibilità, costo, latenza dei giocatori, ecc.) e avviando una sessione di gioco.
- Pool di utenti e client Amazon Cognito per consentire ai giocatori di autenticarsi e iniziare una partita.
- Autorizzatore API Gateway che collega il pool di utenti con APIs.
- Elenco di controllo degli accessi Web (ACL) per limitare le chiamate eccessive dei giocatori all'API Gateway.
- Servizio di backend per effettuare richieste al Amazon GameLift Servers servizio per conto dei client di gioco, ad esempio per richiedere sessioni di gioco e partecipare ai giochi:
  - API Gateway + funzione Lambda per consentire ai giocatori di richiedere uno slot per la sessione di gioco. Questa funzione chiama `StartMatchmaking()` se non sono disponibili slot aperti.
  - API Gateway + funzione Lambda per consentire ai giocatori di ottenere informazioni di connessione per la loro richiesta di gioco.
- Tabelle DynamoDB per archiviare i ticket di matchmaking per i giocatori e le informazioni sulle sessioni di gioco.
- Argomento Amazon SNS + funzione Lambda per gestire gli eventi. GameSessionQueue

Implementa una flotta di container

Quando la configurazione della flotta è completa, scegli il pulsante Implementa la flotta di container per avviare l'implementazione. Questo processo può richiedere diversi minuti mentre il plug-in crea un'immagine del container e la invia a ECR, fornisce le risorse di hosting per la flotta di container, distribuisce la flotta e altre AWS risorse per lo scenario della soluzione di hosting selezionata.

Quando si avvia una distribuzione, è possibile monitorare lo stato di avanzamento di ogni fase. A seconda della configurazione, i passaggi potrebbero includere quanto segue:

- Configurazione dell'immagine del contenitore
- Creazione di un repository Amazon ECR
- Creare un'immagine e passare ad Amazon ECR
- Creazione della definizione di un gruppo di contenitori

- Creazione di una flotta di container

Per informazioni più dettagliate sulla distribuzione, scegli Visualizza nella console AWS di gestione. Quando la flotta di container raggiunge lo stato attivo, utilizza attivamente container con processi server pronti per ospitare sessioni di gioco.

Una volta completata l'implementazione, disponi di una flotta di container funzionante pronta per ospitare sessioni di gioco e accettare connessioni tra giocatori.

Non puoi interrompere una distribuzione in corso. Se la distribuzione entra in uno stato errato o fallisce, puoi ricominciare da capo utilizzando l'opzione Reset deployment.

Avvia il client

A questo punto, hai completato tutte le attività per avviare e giocare al gioco multiplayer ospitato conAmazon GameLift Servers. Per giocare, scegli Start Client per avviare un'istanza locale del tuo client di gioco.

- Se hai utilizzato lo scenario a flotta singola, apri un'istanza del client di gioco con un solo giocatore e accedi alla mappa del server per muoverti. Puoi aprire una seconda istanza del client di gioco per aggiungere un secondo giocatore alla stessa mappa di gioco del server.
- Se hai implementato FlexMatch lo scenario, la soluzione di hosting attende che almeno due client di gioco effettuino richieste di matchmaking. Apri almeno due istanze del tuo client di gioco con un giocatore. I due giocatori verranno abbinati e verrà richiesto di partecipare a una sessione di gioco per la partita.

Aggiorna una flotta di container

Se hai implementato con successo una soluzione di hosting di container gestiti, puoi utilizzare la funzionalità Update deployment. Questa opzione consente di aggiornare le impostazioni di configurazione per una flotta di container distribuita, senza dover creare una nuova flotta.

Quando aggiorni una distribuzione, puoi distribuire un'immagine del contenitore con una build diversa del server di gioco, modificare il repository Amazon ECR, scegliere uno scenario di distribuzione diverso e personalizzare le impostazioni di configurazione opzionali.

Quando sei pronto per distribuire le modifiche, scegli Aggiorna. Il tempo necessario per un aggiornamento della distribuzione è simile a quello di una distribuzione completa. Per informazioni dettagliate sulla distribuzione, scegli Visualizza nella console AWS di gestione.

Pulisci le risorse distribuite

Come best practice, pulisci AWS le risorse per la tua soluzione di container gestiti non appena non ne hai più bisogno. Potresti continuare a sostenere dei costi per queste risorse se non le rimuovi.

Elimina le seguenti risorse:

- Stack di risorse del contenitore gestito. Le risorse in questo stack dipendono dallo scenario di distribuzione selezionato. Per eliminare l'intero stack, usa la AWS CloudFormation console. Gli stack generati dal Amazon GameLift Servers plugin utilizzano la seguente convenzione di denominazione:. `GameLiftPluginForUnreal-{GameName}-Containers` Attendi il completamento del processo di eliminazione dello stack prima di avviare una nuova distribuzione di contenitori gestiti nel plug-in. Per ulteriori informazioni, consulta Eliminare uno stack dalla console. CloudFormation
- Repository Amazon ECR. Se hai utilizzato il plug-in per creare un repository per l'immagine del contenitore, potresti voler eliminare tutti i repository che non sono più necessari. Non è necessario eliminare un repository prima di reimpostare la distribuzione di contenitori gestiti. Se aggiorni o ripristini una distribuzione, il plug-in utilizzerà automaticamente lo stesso repository a meno che non venga richiesto di utilizzarne un altro. Per ulteriori informazioni, consulta Eliminazione di un repository privato in Amazon ECR.

Amazon GameLift Serversplugin per Unity (server SDK 5.x)

Questo plugin aggiunge l'SDK e gli strumenti del server Amazon GameLift Servers C# all'editor UE. Utilizza i flussi di lavoro guidati dell'interfaccia utente per integrare la funzionalità SDK del server nel tuo progetto di gioco e implementare una soluzione di Amazon GameLift Servers hosting per il tuo server di gioco.

Con il plug-in, puoi creare una soluzione di hosting funzionante di base e quindi ottimizzarla e personalizzarla secondo necessità. Configura una flotta Amazon GameLift Servers Anywhere con la tua workstation locale come host. Per l'hosting su cloud con flotte di container gestite EC2 o gestite, implementa il server di gioco con una soluzione completa per gestire le richieste di sessioni di gioco e le connessioni dei client.

Argomenti

- Installa il plugin per il tuo progetto di gioco Unreal
- Plugin per Unity: configura un profilo AWS utente

- Plugin per Unity: configura i test locali con Anywhere Amazon GameLift Servers
- Plugin per Unity: distribuisci il tuo gioco su una flotta gestita EC2
- Plugin per Unity: distribuisci il gioco su una flotta di container gestita

Installa il plugin per il tuo progetto di gioco Unreal

Scarica il Amazon GameLift Servers plugin per Unity da GitHub

Consulta il readme del GitHub repository per informazioni su come installare il plugin per un progetto di gioco.

Il plugin include questi componenti:

- Moduli plug-in per l'editor Unity. Quando il plugin è installato, una nuova voce del menu principale consente di accedere alle Amazon GameLift Servers funzionalità.
- Librerie C# per l'API Amazon GameLift Servers di servizio con funzionalità lato client.
- Librerie C# per l'SDK del Amazon GameLift Servers server (versione 5.x).
- Esempi di contenuti di gioco, tra cui risorse e scene, per poterli provare Amazon GameLift Servers anche se non disponi di un gioco multigiocatore pronto per l'installazione.
- Configurazioni delle soluzioni, fornite come AWS CloudFormation modelli, utilizzate dal plug-in per distribuire il server di gioco sul cloud per l'hosting.

Questo plug-in utilizza AWS CloudFormation modelli per implementare soluzioni di hosting per scenari di gioco comuni. Utilizza queste soluzioni così come fornite o personalizzale secondo necessità per i tuoi giochi.

Plugin per Unity: configura un profilo AWS utente

Dopo aver installato il plugin, configura un profilo utente con un AWS account valido. Puoi mantenere più profili nel plugin, ma puoi selezionare solo un profilo alla volta. Ogni volta che lavori nel plugin, seleziona un profilo da usare. Ogni pagina del flusso di lavoro mostra il profilo attualmente selezionato.

Il mantenimento di più profili consente di passare da una distribuzione di hosting all'altra. Ad esempio, è possibile configurare profili che utilizzano lo stesso AWS account ma vengono distribuiti in regioni diverse. AWS In alternativa, è possibile configurare profili con AWS account o utenti e set di autorizzazioni diversi.

> **Note**
>
> Se hai installato la AWS CLI sulla tua workstation e hai già un profilo configurato, il Amazon GameLift Servers plugin lo rileverà e lo elencherà come profilo esistente. Il plugin seleziona automaticamente qualsiasi profilo denominato `[default]` È possibile utilizzare un profilo esistente o crearne uno nuovo.

Tutti i profili devono essere avviati per configurare alcune AWS risorse necessarie nell'account utente.

Per gestire i tuoi profili AWS

1. Nella barra degli strumenti principale di Unity, scegli il Amazon GameLift Servers menu e seleziona Credenziali di AWS accesso. Questa azione apre il Amazon GameLift Servers plugin alla pagina Configura il tuo profilo utente.
2. Utilizza i pulsanti per creare un nuovo AWS account o configurare un profilo utente per un AWS account che già possiedi.
3. Se non disponi già di un profilo utente, ti verrà richiesto di inserire i dettagli del profilo e creare un nuovo profilo. Inserisci le informazioni che seguono:
   - Un AWS account. Se crei un nuovo AWS account, usa il link AWS Management Console e segui le istruzioni. Vedi Creare un AWS account per maggiori dettagli.
   - Un AWS utente con autorizzazioni all'uso Amazon GameLift Servers e altri AWS servizi richiesti. Consulta Configura un Account AWS le istruzioni sulla configurazione di un utente AWS Identity and Access Management (IAM) con Amazon GameLift Servers autorizzazioni e accesso programmatico con credenziali a lungo termine.
   - Credenziali per il tuo utente. AWS Queste credenziali sono costituite da un ID della chiave di AWS accesso e da una chiave AWS segreta. Per maggiori dettagli, consulta Ottenere le chiavi di accesso.
   - AWS regione. Questa è una posizione geografica in cui desideri creare AWS le tue risorse per l'hosting. Durante lo sviluppo, ti consigliamo di utilizzare un'area vicina alla tua posizione fisica. Scegli una regione dall'elenco delle AWS regioni supportate.
4. Se il plugin rileva profili esistenti, visualizza un elenco di profili disponibili. Seleziona un profilo esistente dall'elenco o scegli Aggiungi un altro profilo per crearne uno nuovo.

Avvia un profilo utente

Tutti i profili devono essere avviati per essere utilizzati con il plugin. Amazon GameLift Servers Il bootstrap crea un bucket Amazon S3 specifico per il profilo. Viene utilizzato per archiviare configurazioni di progetto, creare artefatti e altre dipendenze. I bucket non vengono condivisi tra altri profili.

Il bootstrap implica la creazione di nuove AWS risorse e potrebbe comportare dei costi.

Per avviare il tuo profilo:

1. Nella pagina Credenziali di AWS accesso, controlla lo stato di bootstrap del profilo utente che desideri utilizzare. Se lo stato di bootstrap del profilo è «Inattivo» e non è presente alcun bucket S3 nell'elenco, devi avviare il profilo.
2. Seleziona il profilo che desideri utilizzare e scegli il profilo Bootstrap.
3. Attendi che lo stato del bootstrap passi a «Attivo». Ciò può richiedere alcuni minuti.

Plugin per Unity: configura i test locali con Anywhere Amazon GameLift Servers

In questo flusso di lavoro, aggiungete il codice di gioco del client e del server per verificarne la Amazon GameLift Servers funzionalità e utilizzate il plug-in per designare la workstation locale come host del server di gioco di prova. Una volta completate le attività di integrazione, utilizzate il plug-in per creare i componenti del client e del server di gioco.

Per avviare il flusso di lavoro Amazon GameLift Servers Anywhere:

- Nel menu principale dell'editor di Unity, scegli Amazon GameLift Serverse seleziona Host with Anywhere. Questa azione apre la pagina del plugin per configurare il gioco con una flotta @Anywhere. La pagina presenta un processo in cinque fasi per integrare, creare e lanciare i componenti del gioco.

Imposta il tuo profilo

Scegli il profilo che desideri utilizzare quando segui questo flusso di lavoro. Il profilo selezionato influisce su tutte le fasi del flusso di lavoro. Tutte le risorse create sono associate all' AWS account del profilo e collocate nella AWS regione predefinita del profilo. Le autorizzazioni dell'utente del profilo determinano l'accesso alle AWS risorse e alle azioni.

1. Seleziona un profilo dall'elenco a discesa dei profili disponibili. Se non hai ancora un profilo o desideri crearne uno nuovo, vai al Amazon GameLift Serversmenu e scegli Imposta profili AWS account.
2. Se lo stato di bootstrap non è «Attivo», scegli il profilo Bootstrap e attendi che lo stato diventi «Attivo».

Integra il codice del gioco con l'SDK del server C#

> **Note**
>
> Se hai importato il gioco di esempio, puoi saltare questo passaggio. Le risorse del gioco di esempio dispongono già del codice server e client necessari.

Per questa fase del flusso di lavoro, apporti aggiornamenti al codice client e server del progetto di gioco.

- * I server di gioco devono essere in grado di comunicare con il Amazon GameLift Servers servizio per ricevere richieste di avvio di una sessione di gioco, fornire informazioni sulla connessione alla sessione di gioco e segnalare lo stato.
- I client di gioco devono essere in grado di ottenere informazioni sulle sessioni di gioco, partecipare o avviare sessioni di gioco e ottenere informazioni di connessione per partecipare a una partita.

Integra il codice del tuo server

Se stai usando il tuo progetto di gioco con scene personalizzate, usa il codice di esempio fornito per aggiungere il codice server richiesto al tuo progetto di gioco:

1. Nei file di progetto del gioco, apri la `Assets/Scripts/Server` cartella. Se non esiste, creala.
2. Vai al GitHub repository aws/ amazon-gamelift-plugin-unity e apri il percorso. `Samples~/SampleGame/Assets/Scripts/Server`
3. Individua il file `GameLiftServer.cs` e copialo nella cartella del tuo progetto di gioco. `Server` Quando crei un eseguibile sul server, usa questo file come obiettivo di compilazione.

Il codice di esempio include questi elementi minimi richiesti, che utilizzano l'SDK del server Amazon GameLift Servers C# (versione 5):

- Inizializza un client API. Amazon GameLift Servers La `InitSDK()` chiamata con i parametri del server è necessaria per una flotta Amazon GameLift Servers Anywhere. Queste impostazioni vengono impostate automaticamente per l'uso nel plug-in.
- Implementa le funzioni di callback richieste per rispondere alle richieste del Amazon GameLift Servers servizio, tra cui `OnStartGameSessionOnProcessTerminate`, e. `onHealthCheck`
- Chiamate `ProcessReady()` con una porta designata per avvisare il Amazon GameLift Servers servizio quando il processo del server è pronto per ospitare sessioni di gioco.

Se desideri personalizzare il codice server di esempio, consulta queste risorse:

- Aggiungi Amazon GameLift Servers al tuo server di gioco
- Server C# SDK 5.x per -- Azioni Amazon GameLift Servers

Integra il codice del tuo cliente

Se stai usando il tuo progetto di gioco con scene personalizzate, devi integrare le funzionalità di base nel tuo client di gioco. Devi anche aggiungere elementi dell'interfaccia utente in modo che i giocatori possano accedere e partecipare a una sessione di gioco. Utilizza l'API di servizio per Amazon GameLift Servers (nell' AWS SDK) ottenere informazioni sulla sessione di gioco, creare nuove sessioni di gioco o partecipare a sessioni di gioco esistenti,

Quando crei un client per i test locali con una flotta Anywhere, puoi aggiungere chiamate dirette al Amazon GameLift Servers servizio. Quando sviluppi il tuo gioco per l'hosting su cloud o se prevedi di utilizzare le flotte di Anywhere per l'hosting di produzione, dovrai creare un servizio di backend lato client per gestire tutte le comunicazioni tra i client di gioco e il servizio. Amazon GameLift Servers

Per l'integrazione Amazon GameLift Servers nel codice client, utilizza le seguenti risorse come guida.

- Integra il client con la GameLiftCoreApi classe nel GitHub repository amazon-gamelift-plugin-unity aws/. Questa classe fornisce controlli per l'autenticazione del giocatore e per il recupero delle informazioni sulla sessione di gioco.
- Visualizza esempi di integrazioni di gioco, disponibili nel repository GitHub aws/,. amazon-gamelift-plugin-unity `Samples~/SampleGame/Assets/Scripts/Client/GameLiftClient.cs`
- Segui le istruzioni in Aggiungi al tuo client Amazon GameLift Servers di gioco Unity.

Per i client di gioco che si connettono a una flotta Anywhere, il client di gioco necessita delle seguenti informazioni. Il plug-in aggiorna automaticamente il progetto di gioco per utilizzare le risorse create nel plug-in.

- FleetId - L'identificatore univoco per la tua flotta Anywhere.
- FleetLocation - La posizione personalizzata della tua flotta Anywhere.
- AwsRegion - La AWS regione in cui è ospitata la tua flotta Anywhere. Questa è la regione che hai impostato nel tuo profilo utente.
- ProfileName - Un profilo di AWS credenziali sul computer locale che consente l'accesso all' AWS SDK per. Amazon GameLift Servers Il client di gioco utilizza queste credenziali per autenticare le richieste al servizio. Amazon GameLift Servers

> **Note**
>
> Il profilo delle credenziali viene generato dal plugin e memorizzato sul computer locale. Di conseguenza, è necessario eseguire il client sul computer locale (o su un computer con lo stesso profilo).

Connect a una flotta Anywhere

In questa fase, designerai una flotta Anywhere da utilizzare. Una flotta Anywhere definisce una raccolta di risorse di elaborazione, che possono essere posizionate ovunque, per l'hosting di server di gioco.

- Se l' AWS account che stai utilizzando attualmente dispone di flotte Anywhere esistenti, apri il campo a discesa Fleet name e scegli una flotta. Questo menu a discesa mostra solo le flotte Anywhere nella AWS regione per il profilo utente attualmente attivo.
- Se non ci sono flotte esistenti o desideri crearne una nuova, scegli Crea una nuova flotta Anywhere e fornisci un nome per la flotta.

Dopo aver scelto una flotta Anywhere per il tuo progetto, Amazon GameLift Servers verifica che lo stato della flotta sia attivo e visualizza l'ID della flotta. Puoi tenere traccia dello stato di avanzamento di questa richiesta nel registro di output dell'editor Unity.

Registra un calcolo

In questo passaggio, registri la tua workstation locale come risorsa di elaborazione nella nuova flotta Anywhere.

1. Immettete un nome di calcolo per il computer locale. Se aggiungi più di un computer nel parco dati, i nomi devono essere univoci.
2. Scegli Register compute. Puoi tenere traccia dello stato di avanzamento di questa richiesta nel registro di output dell'editor Unity.

Il plugin registra la workstation locale con l'indirizzo IP impostato su localhost (127.0.0.1). Questa impostazione presuppone che il client e il server di gioco vengano eseguiti sulla stessa macchina.

In risposta a questa azione, Amazon GameLift Servers verifica che sia in grado di connettersi al computer e restituisce informazioni sull'elaborazione appena registrata.

Avvia il gioco

In questo passaggio crei i componenti del gioco e li avvii per giocare. Completa le seguenti operazioni:

1. Configura il tuo client di gioco. In questo passaggio, richiedi al plugin di aggiornare una `GameLiftClientSettings` risorsa per il tuo progetto di gioco. Il plugin utilizza questa risorsa per memorizzare determinate informazioni di cui il client di gioco ha bisogno per connettersi al Amazon GameLift Servers servizio.

    a. Se non hai importato e inizializzato il gioco di esempio, crea una nuova `GameLiftClientSettings` risorsa. Nel menu principale dell'editor Unity, scegli Assets, Create, Amazon GameLift, Client Settings. Se crei più copie del `GameLiftClientSettings` tuo progetto, il plug-in lo rileva automaticamente e ti notifica quale risorsa verrà aggiornata dal plug-in.

    b. In Launch Game, scegli Configure Client: Apply Anywhere Settings. Questa azione aggiorna le impostazioni del client di gioco per utilizzare la flotta Anywhere che hai appena configurato.
2. Crea ed esegui il tuo client di gioco.

    a. Crea un eseguibile client utilizzando il processo di compilazione standard di Unity. In File, Build Settings, passa la piattaforma a Windows, Mac, Linux. Se hai importato il gioco di

esempio e hai inizializzato le impostazioni, l'elenco delle build e l'obiettivo di build vengono aggiornati automaticamente.

b. Avvia una o più istanze dell'eseguibile del client di gioco appena creato.

3. Avvia un server di gioco nella tua flotta Anywhere. Scegli Server: Avvia il server nell'editor. Questa attività avvia un server live a cui il client può connettersi finché l'editor Unity rimane aperto.
4. Avvia o partecipa a una sessione di gioco. Nelle istanze del client di gioco, usa l'interfaccia utente per unire ogni client a una sessione di gioco. Il modo in cui esegui questa operazione dipende da come hai aggiunto funzionalità al client.

Se utilizzi il client di gioco di esempio, ha le seguenti caratteristiche:

- Un componente per il login del giocatore. Quando ci si connette a un server di gioco su una flotta Anywhere, non è prevista la convalida del giocatore. Puoi inserire qualsiasi valore per partecipare alla sessione di gioco.
- Una semplice interfaccia utente di accesso al gioco. Quando un client tenta di partecipare a una partita, cerca automaticamente una sessione di gioco attiva con uno slot disponibile. Se non è disponibile alcuna sessione di gioco, il client richiede una nuova sessione di gioco. Se è disponibile una sessione di gioco, il client richiede di partecipare alla sessione di gioco disponibile. Quando provi il gioco con più client simultanei, il primo client avvia la sessione di gioco e i client rimanenti si uniscono automaticamente alla sessione di gioco esistente.
- Sessioni di gioco con quattro slot per giocatori. Puoi avviare fino a quattro istanze del client di gioco contemporaneamente e queste si uniranno alla stessa sessione di gioco.

Avvio da un server eseguibile (opzionale)

Puoi creare e avviare il tuo server di gioco eseguibile per testarlo su una flotta Anywhere.

1. Crea un eseguibile sul server utilizzando il processo di compilazione standard di Unity. In File, Build Settings, passa la piattaforma a Server dedicato e crea.
2. Ottieni un token di autenticazione a breve termine chiamando il comando AWS CLI get-compute-auth-tokencon l'ID e AWS la regione della flotta Anywhere. L'ID della flotta viene visualizzato in Connect to an Anywhere Fleet quando crei la flotta. La AWS regione viene visualizzata in Imposta il tuo profilo quando selezioni il tuo profilo attivo.

```
aws gamelift get-compute-auth-token --fleet-id [your anywhere fleet ID] --region
 [your AWS region]
```

3. Avvia l'eseguibile del server di gioco appena creato da una riga di comando e inserisci un token di autenticazione valido.

```
my_project.exe --authToken [token]
```

Plugin per Unity: distribuisci il tuo gioco su una flotta gestita EC2

In questo flusso di lavoro, utilizzi il plug-in per preparare il gioco per l'hosting su risorse di elaborazione basate su cloud gestite da. Amazon GameLift Servers Aggiungi il codice di gioco del client e del server per Amazon GameLift Servers aumentarne la funzionalità, quindi carica la build del server sul Amazon GameLift Servers servizio per l'hosting. Una volta completato questo flusso di lavoro, avrai server di gioco in esecuzione nel cloud e un client di gioco funzionante in grado di connettersi ad essi.

Per avviare il EC2 flusso di lavoro Amazon GameLift Servers gestito di Amazon:

- Nel menu principale dell'editor Unity, scegli Amazon GameLift Serverse seleziona Host with Managed EC2. Questo flusso di lavoro presenta un processo in sei fasi per integrare, creare, distribuire e avviare i componenti del gioco.

Imposta il tuo profilo

Scegli il profilo che desideri utilizzare quando segui questo flusso di lavoro. Il profilo selezionato influisce su tutte le fasi del flusso di lavoro. Tutte le risorse create sono associate all' AWS account del profilo e collocate nella AWS regione predefinita del profilo. Le autorizzazioni dell'utente del profilo determinano l'accesso alle AWS risorse e alle azioni.

1. Seleziona un profilo dall'elenco a discesa dei profili disponibili. Se non hai ancora un profilo o desideri crearne uno nuovo, vai al Amazon GameLift Servers menu e scegli Imposta profili AWS account.
2. Se lo stato di bootstrap non è «Attivo», scegli il profilo Bootstrap e attendi che lo stato diventi «Attivo».

Integra il tuo gioco con Amazon GameLift Servers

Per questo compito, aggiorni il codice del client e del server nel tuo progetto di gioco.

- I server di gioco devono essere in grado di comunicare con il Amazon GameLift Servers servizio per ricevere richieste di avvio di una sessione di gioco, fornire informazioni sulla connessione alla sessione di gioco e segnalare lo stato.
- I client di gioco devono essere in grado di ottenere informazioni sulle sessioni di gioco, partecipare o avviare sessioni di gioco e ottenere informazioni di connessione per partecipare a una partita.

> **Note**
>
> Se hai importato il gioco di esempio, puoi saltare questo passaggio. Le risorse del gioco di esempio dispongono già del codice server e client necessari.

Integra il codice del tuo server

Quando utilizzi il tuo progetto di gioco con scene personalizzate, usa il codice di esempio fornito per aggiungere il codice server richiesto al tuo progetto di gioco. Se hai integrato il tuo progetto di gioco per testarlo con una flotta Anywhere, hai già completato le istruzioni in questo passaggio.

1. Nei file di progetto del gioco, apri la `Assets/Scripts/Server` cartella. Se non esiste, creala.
2. Vai al GitHub repository aws/ amazon-gamelift-plugin-unity e apri il percorso. `Samples~/SampleGame/Assets/Scripts/Server`
3. Individua il file `GameLiftServer.cs` e copialo nella cartella del tuo progetto di gioco. `Server` Quando crei un eseguibile sul server, usa questo file come obiettivo di compilazione.

Il codice di esempio include questi elementi minimi richiesti, che utilizzano l'SDK del server Amazon GameLift Servers C# (versione 5):

- Inizializza un client API. Amazon GameLift Servers La chiamata initSDK () con i parametri del server è necessaria per Amazon GameLift Servers una flotta Anywhere. Queste impostazioni vengono impostate automaticamente per l'uso nel plug-in.
- Implementa le funzioni di callback richieste per rispondere alle richieste del Amazon GameLift Servers servizio, tra cui `OnStartGameSession` `OnProcessTerminate`, e. `onHealthCheck`

- Chiamate `ProcessReady()` con una porta designata per avvisare il Amazon GameLift Servers servizio quando il processo del server è pronto per ospitare sessioni di gioco.

Se desideri personalizzare il codice server di esempio, consulta queste risorse:

- Aggiungi Amazon GameLift Servers al tuo server di gioco
- Server C# SDK 5.x per -- Azioni Amazon GameLift Servers

Integra il codice del tuo cliente

Per i client di gioco che si connettono a server di gioco basati su cloud, è consigliabile utilizzare un servizio di backend lato client per effettuare chiamate al Amazon GameLift Servers servizio, anziché effettuare le chiamate direttamente dal client di gioco.

Nel flusso di lavoro dei plug-in per l'hosting su una EC2 flotta gestita, ogni scenario di implementazione include un servizio di backend predefinito che include i seguenti componenti:

- Un set di funzioni Lambda e tabelle DynamoDB utilizzate per richiedere sessioni di gioco e recuperare informazioni sulla sessione di gioco. Questi componenti utilizzano un gateway API come proxy.
- Un pool di utenti Amazon Cognito che genera giocatori unici IDs e autentica le connessioni dei giocatori.

Per utilizzare questi componenti, il client di gioco necessita della funzionalità necessaria per inviare richieste al servizio di backend per effettuare le seguenti operazioni:

- Crea un utente giocatore nel pool di utenti di AWS Cognito ed esegui l'autenticazione.
- Partecipa a una sessione di gioco e ricevi informazioni sulla connessione.
- Partecipa a una partita usando matchmaking.

Utilizza le seguenti risorse come guida.

- Integra il client con la GameLiftCoreApiclasse nel GitHub repository amazon-gamelift-plugin-unityaws/. Questa classe fornisce controlli per l'autenticazione del giocatore e per il recupero delle informazioni sulla sessione di gioco.

- Per visualizzare le integrazioni di gioco di esempio, vai al repository aws/ GitHub ,. amazon-gamelift-plugin-unity `Samples~/SampleGame/Assets/Scripts/Client/GameLiftClient.cs`
- Aggiungi Amazon GameLift Servers al tuo client di gioco.

Seleziona lo scenario di implementazione

In questo passaggio, scegli la soluzione di hosting di giochi che desideri implementare in questo momento. Puoi avere più distribuzioni del gioco, utilizzando uno qualsiasi degli scenari.

- Flotta a regione singola: distribuisce il server di gioco su un'unica flotta di risorse di hosting nella regione predefinita del profilo attivo. AWS Questo scenario è un buon punto di partenza per testare l'integrazione del server AWS e la configurazione della build del server. Implementa le seguenti risorse:
  - AWS fleet (On-Demand) con la build del server di gioco installata e funzionante.
  - Pool di utenti e client Amazon Cognito per consentire ai giocatori di autenticarsi e iniziare una partita.
  - Autorizzatore di gateway API che collega il pool di utenti con. APIs
  - Web ACl per limitare le chiamate eccessive dei giocatori al gateway API.
  - API gateway + funzione Lambda per consentire ai giocatori di richiedere uno slot di gioco. Questa funzione chiama `CreateGameSession()` se non ce ne sono disponibili.
  - API gateway + funzione Lambda per consentire ai giocatori di ottenere informazioni di connessione per la loro richiesta di gioco.
- FlexMatch flotta: distribuisce il tuo server di gioco su una serie di flotte e configura un FlexMatch matchmaker con le regole per creare partite tra giocatori. Questo scenario utilizza un hosting Spot a basso costo con una struttura multi-flotta e più sedi per una disponibilità duratura. Questo approccio è utile quando sei pronto per iniziare a progettare un componente matchmaker per la tua soluzione di hosting. In questo scenario, creerai le risorse di base per questa soluzione, che potrai personalizzare in seguito, se necessario. Implementa le seguenti risorse:
  - FlexMatch configurazione del matchmaking e regole di matchmaking impostate per accettare le richieste dei giocatori e formare partite.
  - Tre AWS flotte con la configurazione del server di gioco installata e funzionante in più località. Include due flotte Spot e una flotta On-Demand come backup.

- AWS coda per il posizionamento delle sessioni di gioco che soddisfa le richieste di partite proposte trovando la migliore risorsa di hosting possibile (in base a fattibilità, costo, latenza dei giocatori, ecc.) e avviando una sessione di gioco.
- Pool di utenti e client Amazon Cognito per consentire ai giocatori di autenticarsi e iniziare una partita.
- Autorizzatore di gateway API che collega il pool di utenti con. APIs
- Web ACl per limitare le chiamate eccessive dei giocatori al gateway API.
- API gateway + funzione Lambda per consentire ai giocatori di richiedere uno slot di gioco. Questa funzione chiama`StartMatchmaking()`.
- API gateway + funzione Lambda per consentire ai giocatori di ottenere informazioni di connessione per la loro richiesta di gioco.
- Tabelle Amazon DynamoDB per archiviare i ticket di matchmaking per i giocatori e le informazioni sulle sessioni di gioco.
- Argomento SNS + funzione Lambda per GameSessionQueue gestire gli eventi.

Imposta i parametri di gioco

In questo passaggio, descrivi il gioco su cui caricarlo AWS.

- Nome del gioco: fornisci un nome significativo per il tuo progetto di gioco. Questo nome viene utilizzato all'interno del plugin.
- Nome della flotta: fornisci un nome significativo per il tuo EC2 parco veicoli gestito. Amazon GameLift Serversutilizza questo nome (insieme all'ID della flotta) per elencare le risorse nella AWS console.
- Nome build: fornisci un nome significativo per la build del server. AWS utilizza questo nome per fare riferimento alla copia della build del server che viene caricata Amazon GameLift Servers e utilizzata per le distribuzioni.
- Parametri di avvio: inserisci le istruzioni opzionali da eseguire all'avvio del server eseguibile su un'istanza gestita EC2 del parco istanze. La lunghezza massima è di 1024 caratteri.
- Cartella del server di gioco: fornisci il percorso di una cartella locale contenente la build del server.
- File del server di gioco: specifica il nome del file eseguibile del server.

Scenario di distribuzione

In questo passaggio, distribuisci il gioco su una soluzione di hosting cloud in base allo scenario di implementazione scelto. Questo processo può richiedere diversi minuti per AWS la convalida della build del server, il provisioning delle risorse di hosting, l'installazione del server di gioco, l'avvio dei processi del server e la preparazione per ospitare sessioni di gioco.

Per iniziare la distribuzione, scegli Deploy. CloudFormation Puoi monitorare lo stato del tuo hosting di giochi qui. Per informazioni più dettagliate, puoi accedere alla console di AWS gestione AWS e visualizzare le notifiche degli eventi. Assicurati di accedere utilizzando lo stesso account, utente e AWS regione del profilo utente attivo nel plug-in.

Una volta completata la distribuzione, il server di gioco è installato su un' AWS EC2 istanza. Almeno un processo del server è in esecuzione ed è pronto per iniziare una sessione di gioco.

Avvia il client di gioco

Quando la tua flotta sarà dispiegata con successo, ora avrai dei server di gioco attivi e disponibili per ospitare sessioni di gioco. Ora puoi creare il tuo client, avviarlo, connetterti per partecipare alla sessione di gioco.

1. Configura il tuo client di gioco. In questo passaggio, richiedi al plugin di aggiornare una `GameLiftClientSettings` risorsa per il tuo progetto di gioco. Il plugin utilizza questa risorsa per memorizzare determinate informazioni di cui il client di gioco ha bisogno per connettersi al Amazon GameLift Servers servizio.

    a. Se non hai importato e inizializzato il gioco di esempio, crea una nuova `GameLiftClientSettings` risorsa. Nel menu principale dell'editor Unity, scegli Assets, Create, Amazon GameLift, Client Settings. Se crei più copie del `GameLiftClientSettings` tuo progetto, il plug-in lo rileva automaticamente e ti notifica quale risorsa verrà aggiornata dal plug-in.

    b. In Launch Game, scegli Configura client: applica impostazioni gestite EC2. Questa azione aggiorna le impostazioni del client di gioco per utilizzare la EC2 flotta gestita che hai appena schierato.

2. Crea il tuo client di gioco. Crea un eseguibile client utilizzando il processo di compilazione standard di Unity. In File, Build Settings, passa la piattaforma a Windows, Mac, Linux. Se hai importato il gioco di esempio e hai inizializzato le impostazioni, l'elenco delle build e l'obiettivo di build vengono aggiornati automaticamente.

3. Avvia l'eseguibile del client di gioco di nuova build. Per iniziare a giocare, avvia da due a quattro istanze client e usa l'interfaccia utente di ciascuna per partecipare a una sessione di gioco.

Se utilizzi il client di gioco di esempio, ha le seguenti caratteristiche:

- Un componente per il login del giocatore. Quando ci si connette a un server di gioco su una flotta Anywhere, non è prevista la convalida del giocatore. Puoi inserire qualsiasi valore per partecipare alla sessione di gioco.
- Una semplice interfaccia utente di accesso al gioco. Quando un client tenta di partecipare a una partita, cerca automaticamente una sessione di gioco attiva con uno slot disponibile. Se non è disponibile alcuna sessione di gioco, il client richiede una nuova sessione di gioco. Se è disponibile una sessione di gioco, il client richiede di partecipare alla sessione di gioco disponibile. Quando provi il gioco con più client simultanei, il primo client avvia la sessione di gioco e i client rimanenti si uniscono automaticamente alla sessione di gioco esistente.
- Sessioni di gioco con quattro slot per giocatori. Puoi avviare fino a quattro istanze del client di gioco contemporaneamente e queste si uniranno alla stessa sessione di gioco.

Plugin per Unity: distribuisci il gioco su una flotta di container gestita

Usa questo flusso di lavoro guidato dai plug-in per creare un'immagine del contenitore per il tuo server di gioco e distribuirla su una soluzione di hosting basata su container. Una volta completato con successo questo flusso di lavoro, il server di gioco containerizzato è in esecuzione nel cloud e puoi utilizzare il plug-in per avviare un client di gioco, connetterti a una sessione di gioco e giocare.

Prima di iniziare

Questo flusso di lavoro presuppone che tu abbia completato le seguenti attività.

- Integra il codice del server di gioco con l'SDK Amazon GameLift Servers del server. il nostro server di gioco ospitato deve essere in grado di comunicare con il Amazon GameLift Servers servizio in modo che possa rispondere alle richieste di avvio di nuove sessioni di gioco e segnalare lo stato della sessione di gioco. Se non hai completato questa operazione, ti consigliamo di seguire prima il flusso di lavoro del plug-in Host with Anywhere. Per istruzioni sulla preparazione del codice del server di gioco, consultaIntegra il codice del tuo server. Per una flotta di container gestita, devi integrare il gioco con la versione 5.2 o successiva dell'SDK del server.

> **Note**
>
> Se utilizzi la mappa di gioco di avvio, questa operazione è già stata eseguita per te.

- Package del tuo file eseguibile del server di gioco per eseguirlo su Linux.
- Raccogli i file da distribuire con la build del tuo server di gioco. Sul computer locale, crea una directory di lavoro per organizzare i file, che verrà incorporata nell'immagine del contenitore del server di gioco. Questi potrebbero includere dipendenze di gioco, uno script per avviare i server di gioco e altri processi all'avvio di un contenitore, ecc.
- Integra il codice del client di gioco conAmazon GameLift Servers. Un modo per completare questa operazione è aggiungere una risorsa di esempio (inclusa nel plug-in) già integrata. Per indicazioni sulla preparazione del codice client di gioco, consultaIntegra il codice del tuo cliente.
- Installa Docker sul tuo computer locale. È necessario installare questo strumento se desideri che il plug-in crei immagini dei contenitori per te e le invii a un repository ECR. In alternativa, puoi eseguire queste attività manualmente e indicare al plug-in di utilizzare un'immagine del contenitore esistente. Per ulteriori informazioni sulla creazione manuale dell'immagine, consulta Creare un'immagine del contenitore per Amazon GameLift Servers.

Per avviare il flusso di lavoro dei contenitori Amazon GameLift Servers gestiti:

- Nella barra degli strumenti principale dell'editor Unity, scegli il Amazon GameLift Servers menu e seleziona Contenitori gestiti. Questa azione apre la pagina del plug-in Host with Managed Containers, che presenta un step-by-step processo per creare un'immagine del contenitore con la build del server di gioco, distribuirla in una flotta di container e avviare il gioco.

Passaggio 0: imposta il tuo profilo

Questa sezione mostra il profilo utente attualmente selezionato. Verifica che il profilo utente corrente sia quello che desideri utilizzare per questo flusso di lavoro. Tutte le risorse create in questo flusso di lavoro sono associate all' AWS account del profilo e collocate nella AWS regione predefinita del profilo. Le autorizzazioni dell'utente del profilo determinano l'accesso alle AWS risorse e alle azioni.

Potrebbe essere necessario modificare il profilo utente selezionato se:

- Al momento non è selezionato alcun profilo.
- Vuoi selezionare un profilo diverso o creare un nuovo profilo.

- È necessario eseguire il bootstrap del profilo selezionato (se lo stato di bootstrap è inattivo).

Per impostare o modificare il profilo utente selezionato

- Nel Amazon GameLift Servers menu, scegli Open AWS Access Credentials.

Fase 1: Valutare la disponibilità del contenitore

Prima di distribuire il tuo server di gioco in una flotta di container, devi impacchettarlo in un'immagine del contenitore e archiviarlo in un repository Amazon ECR. Il plug-in può completare queste attività per te oppure puoi eseguirle manualmente. In questo passaggio, fornisci informazioni sullo stato dell'immagine del contenitore e dell'archivio ECR.

Usa le domande di valutazione per indicare al plugin quali passi deve compiere:

- Crea una nuova immagine del contenitore. Se scegli questa opzione, nel passaggio successivo ti verrà richiesta la posizione della directory di build del server di gioco e l'eseguibile della build. Il plugin utilizza un modello Dockerfile (fornito daAmazon GameLift Servers) e lo configura automaticamente per il gioco. Puoi visualizzare il modello all'indirizzo. Crea un'immagine del contenitore per Amazon GameLift Servers Dopo aver scelto questa opzione, indica dove desideri che il plugin memorizzi la nuova immagine:
  - Crea un nuovo repository Amazon ECR e inviaci l'immagine del contenitore. Il plug-in crea un repository ECR privato utilizzando l' AWS account e lo utilizza come impostazione predefinita Regione AWS nel profilo utente selezionato.
  - Invia l'immagine del contenitore a un repository Amazon ECR creato in precedenza. Se scegli questa opzione, il passaggio successivo ti chiederà di selezionare un repository Amazon ECR esistente da un elenco. L'elenco include tutti i repository Amazon ECR per l' AWS account e quelli predefiniti Regione AWS nel profilo utente selezionato. Puoi selezionare un repository pubblico o privato.
- Usa un'immagine del contenitore esistente. Se hai creato un'immagine manualmente, ti consigliamo di utilizzare il modello Dockerfile fornito daAmazon GameLift Servers, disponibile all'indirizzo. Crea un'immagine del contenitore per Amazon GameLift Servers Dopo aver scelto questa opzione, indica dove si trova l'immagine.
  - Un'immagine generata da Docker archiviata localmente. Se scegli questa opzione, il plug-in crea un nuovo repository privato Amazon ECR e vi invia il file di immagine locale. Il passaggio successivo richiederà un ID immagine, che il plug-in utilizza per localizzare il file di immagine.

- Un'immagine del contenitore già archiviata in un repository Amazon ECR. Se scegli questa opzione, il passaggio successivo ti chiederà di selezionare un repository e un'immagine Amazon ECR esistenti da un elenco. L'elenco include tutti i repository Amazon ECR per l' AWS account e quelli predefiniti Regione AWS nel profilo utente selezionato. Puoi selezionare un repository pubblico o privato.

Fase 2: Configurare la distribuzione delle immagini

In questo passaggio, fornisci le informazioni necessarie al plug-in per distribuire l'immagine del container in una flotta di container. Questo passaggio richiede le seguenti informazioni:

- La posizione della build del server di gioco, dell'immagine del contenitore o del repository Amazon ECR, in base alle selezioni effettuate nella Fase 1.
- Lo scenario da utilizzare per la distribuzione dei container gestiti.
- Impostazioni di distribuzione opzionali. Questa sezione contiene le impostazioni di configurazione che il plugin utilizza per impostazione predefinita. È possibile modificarli o mantenere i valori predefiniti
  - Per impostazione predefinita, il nome del gioco è impostato sul nome del progetto di gioco. Tutte le AWS risorse create dal plugin fanno riferimento al valore del nome del gioco.
  - L'intervallo di porte, il limite di memoria e il limite di vCPU sono impostazioni di configurazione per la flotta di container. Per ulteriori informazioni sulla personalizzazione di questi valori, consulta la sezione Configurare le connessioni di rete dedicata all'intervallo delle porte di connessione e Imposta i limiti delle risorse ai limiti delle risorse.
  - Il tag dell'immagine del contenitore viene utilizzato per classificare le immagini del contenitore in Amazon ECR. Il valore predefinito è `unity-gamelift-plugin`.

Opzioni dello scenario di implementazione

Flotta di container in un'unica regione

Questo scenario distribuisce il tuo server di gioco su una singola flotta di container. È un buon punto di partenza per testare l'integrazione del server AWS e la configurazione del container. Implementa le seguenti risorse.

- Amazon GameLift ServersLa definizione del gruppo di container descrive come distribuire ed eseguire le immagini dei container su una flotta di container.

- Amazon GameLift Serversflotta di container (On-Demand) con il container del server di gioco installato e funzionante, con alias.
- Pool di utenti e client Amazon Cognito per consentire ai giocatori di autenticarsi e iniziare una partita.
- Autorizzatore API Gateway che collega il pool di utenti con APIs.
- Elenco di controllo degli accessi Web (ACL) per limitare le chiamate eccessive dei giocatori all'API Gateway.
- Servizio di backend per effettuare richieste al Amazon GameLift Servers servizio per conto dei client di gioco, ad esempio per richiedere sessioni di gioco e partecipare ai giochi:
  - API Gateway + funzione Lambda per consentire ai giocatori di richiedere uno slot per la sessione di gioco. Questa funzione chiama `CreateGameSession()` se non sono disponibili slot aperti.
  - API Gateway + funzione Lambda per consentire ai giocatori di ottenere informazioni di connessione per la loro richiesta di gioco.

Flotta di container a regione singola con FlexMatch

Questo scenario distribuisce il server di gioco su una flotta di container, configura il posizionamento delle sessioni di gioco, e imposta il matchmaking. FlexMatch Questo scenario è utile quando sei pronto per iniziare a progettare un matchmaker personalizzato per la tua soluzione di hosting. Utilizza questo scenario per creare le risorse di base per questa soluzione, che potrai personalizzare in seguito, se necessario. Implementa le seguenti risorse:

- Amazon GameLift Serversdefinizione del gruppo di container che descrive come distribuire ed eseguire le immagini dei container su una flotta di container.
- Amazon GameLift Serversflotta di container (On-Demand) con il container del server di gioco installato e funzionante, con alias.
- FlexMatchconfigurazione del matchmaking e regole di matchmaking impostate per accettare le richieste dei giocatori e formare partite.
- Amazon GameLift Serverscoda di sessione di gioco che soddisfa le richieste di partite proposte trovando la migliore risorsa di hosting possibile (in base a fattibilità, costo, latenza dei giocatori, ecc.) e avviando una sessione di gioco.
- Pool di utenti e client Amazon Cognito per consentire ai giocatori di autenticarsi e iniziare una partita.
- Autorizzatore API Gateway che collega il pool di utenti con APIs.

- Elenco di controllo degli accessi Web (ACL) per limitare le chiamate eccessive dei giocatori all'API Gateway.
- Servizio di backend per effettuare richieste al Amazon GameLift Servers servizio per conto dei client di gioco, ad esempio per richiedere sessioni di gioco e partecipare ai giochi:
  - API Gateway + funzione Lambda per consentire ai giocatori di richiedere uno slot per la sessione di gioco. Questa funzione chiama `StartMatchmaking()` se non sono disponibili slot aperti.
  - API Gateway + funzione Lambda per consentire ai giocatori di ottenere informazioni di connessione per la loro richiesta di gioco.
- Tabelle DynamoDB per archiviare i ticket di matchmaking per i giocatori e le informazioni sulle sessioni di gioco.
- Argomento Amazon SNS + funzione Lambda per gestire gli eventi. GameSessionQueue

Implementa una flotta di container

Quando la configurazione della flotta è completa, scegli il pulsante Implementa la flotta di container per avviare l'implementazione. Questo processo può richiedere diversi minuti mentre il plug-in crea un'immagine del container e la invia a ECR, fornisce le risorse di hosting per la flotta di container, distribuisce la flotta e altre AWS risorse per lo scenario della soluzione di hosting selezionata.

Quando si avvia una distribuzione, è possibile monitorare lo stato di avanzamento di ogni fase. A seconda della configurazione, i passaggi potrebbero includere quanto segue:

- Configurazione dell'immagine del contenitore
- Creazione di un repository Amazon ECR
- Creare un'immagine e passare ad Amazon ECR
- Creazione della definizione di un gruppo di contenitori
- Creazione di una flotta di container

Per informazioni più dettagliate sulla distribuzione, scegli Visualizza nella console AWS di gestione. Quando la flotta di container raggiunge lo stato attivo, utilizza attivamente container con processi server pronti per ospitare sessioni di gioco.

Una volta completata l'implementazione, disponi di una flotta di container funzionante pronta per ospitare sessioni di gioco e accettare connessioni tra giocatori.

Non puoi interrompere una distribuzione in corso. Se la distribuzione entra in uno stato errato o fallisce, puoi ricominciare da capo utilizzando l'opzione Reset deployment.

Avvia il client

A questo punto, hai completato tutte le attività per avviare e giocare al gioco multiplayer ospitato conAmazon GameLift Servers. Per giocare, scegli Start Client per avviare un'istanza locale del tuo client di gioco.

- Se hai utilizzato lo scenario a flotta singola, apri un'istanza del client di gioco con un solo giocatore e accedi alla mappa del server per muoverti. Puoi aprire una seconda istanza del client di gioco per aggiungere un secondo giocatore alla stessa mappa di gioco del server.
- Se hai implementato FlexMatch lo scenario, la soluzione di hosting attende che almeno due client di gioco effettuino richieste di matchmaking. Apri almeno due istanze del tuo client di gioco con un giocatore. I due giocatori verranno abbinati e verrà richiesto di partecipare a una sessione di gioco per la partita.

Aggiorna una flotta di container

Se hai implementato con successo una soluzione di hosting di container gestiti, puoi utilizzare la funzionalità Update deployment. Questa opzione consente di aggiornare le impostazioni di configurazione per una flotta di container distribuita, senza dover creare una nuova flotta.

Quando aggiorni una distribuzione, puoi distribuire un'immagine del contenitore con una build diversa del server di gioco, modificare il repository Amazon ECR, scegliere uno scenario di distribuzione diverso e personalizzare le impostazioni di configurazione opzionali.

Quando sei pronto per distribuire le modifiche, scegli Aggiorna. Il tempo necessario per un aggiornamento della distribuzione è simile a quello di una distribuzione completa. Per informazioni dettagliate sulla distribuzione, scegli Visualizza nella console AWS di gestione.

Pulisci le risorse distribuite

Come best practice, pulisci AWS le risorse per la tua soluzione di container gestiti non appena non ne hai più bisogno. Potresti continuare a sostenere dei costi per queste risorse se non le rimuovi.

Elimina le seguenti risorse:

- Stack di risorse del contenitore gestito. Le risorse in questo stack dipendono dallo scenario di distribuzione selezionato. Per eliminare l'intero stack, usa la AWS CloudFormation console.

Gli stack generati dal Amazon GameLift Servers plugin utilizzano la seguente convenzione di denominazione:. `GameLiftPluginForUnity-{GameName}-Containers` Attendi il completamento del processo di eliminazione dello stack prima di avviare una nuova distribuzione di contenitori gestiti nel plug-in. Per ulteriori informazioni, consulta Eliminare uno stack dalla console. CloudFormation

- Repository Amazon ECR. Se hai utilizzato il plug-in per creare un repository per l'immagine del contenitore, potresti voler eliminare tutti i repository che non sono più necessari. Non è necessario eliminare un repository prima di reimpostare la distribuzione di contenitori gestiti. Se aggiorni o ripristini una distribuzione, il plug-in utilizzerà automaticamente lo stesso repository a meno che non venga richiesto di utilizzarne un altro. Per ulteriori informazioni, consulta Eliminazione di un repository privato in Amazon ECR.

Amazon GameLift Serversplugin per Unity (server SDK 4.x)

> **Note**
>
> Questo argomento fornisce informazioni su una versione precedente del Amazon GameLift Servers plug-in per Unity. La versione 1.0.0 (rilasciata nel 2021) utilizza l'SDK del server per Amazon GameLift Servers 4.x o versioni precedenti. Per la documentazione sulla versione più recente del plug-in, che utilizza il server SDK 5.x e supporta Anywhere, consulta. Amazon GameLift Servers Amazon GameLift Serversplugin per Unity (server SDK 5.x)

Amazon GameLift Serversfornisce strumenti per preparare i server di gioco multiplayer su cui funzionare. Amazon GameLift Servers Il Amazon GameLift Servers plug-in per Unity semplifica l'Amazon GameLift Serversintegrazione nei tuoi progetti di gioco Unity e la distribuzione di Amazon GameLift Servers risorse per l'hosting su cloud. Usa il plug-in per Unity per accedere Amazon GameLift Servers APIs e distribuire AWS CloudFormation modelli per scenari di gioco comuni.

Dopo aver configurato il plug-in, puoi provare l'esempio di Amazon GameLift Servers Unity su GitHub.

Argomenti

- Integrazione Amazon GameLift Servers con un progetto di server di gioco Unity
- Integrazione Amazon GameLift Servers con un progetto client di gioco Unity
- Installa e configura il plugin

- Prova il gioco localmente
- Implementa uno scenario
- Integra i giochi con Unity Amazon GameLift Servers
- Importa ed esegui un gioco di esempio

Integrazione Amazon GameLift Servers con un progetto di server di gioco Unity

> **Note**
>
> Questo argomento fornisce informazioni su una versione precedente del Amazon GameLift Servers plug-in per Unity. La versione 1.0.0 (rilasciata nel 2021) utilizza l'SDK del server per Amazon GameLift Servers 4.x o versioni precedenti. Per la documentazione sull'ultima versione del plug-in, che utilizza il server SDK 5.x e supporta Anywhere, consulta. Amazon GameLift Servers Amazon GameLift Serversplugin per Unity (server SDK 5.x)

Questo argomento ti aiuta a preparare il tuo server di gioco personalizzato su cui ospitare. Amazon GameLift Servers Il server di gioco deve essere in grado Amazon GameLift Servers di notificare il suo stato, di avviare e interrompere le sessioni di gioco quando richiesto e di eseguire altre attività. Per ulteriori informazioni, consulta Aggiungi Amazon GameLift Servers al tuo server di gioco.

Prerequisiti

Prima di integrare il server di gioco, completa le seguenti attività:

- Configurare un ruolo di servizio IAM per Amazon GameLift Servers
- Installa e configura il plugin

Configura un nuovo processo del server

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Imposta la comunicazione con Amazon GameLift Servers e segnala che il processo del server è pronto per ospitare una sessione di gioco.

1. Inizializza l'SDK del server chiamando. `InitSDK()`
2. Per preparare il server ad accettare una sessione di gioco, chiama `ProcessReady()` indicando la porta di connessione e i dettagli sulla posizione della sessione di gioco. Includi i nomi delle funzioni di callback richiamate dal Amazon GameLift Servers servizio, ad esempio,`OnGameSession()`,`OnGameSessionUpdate()`. `OnProcessTerminate()` `OnHealthCheck()` Amazon GameLift Serverspotrebbero essere necessari alcuni minuti per fornire una richiamata.
3. Amazon GameLift Serversaggiorna lo stato del processo del server su`ACTIVE`.
4. Amazon GameLift Serverschiama on`HealthCheck` periodicamente.

Il seguente esempio di codice mostra come configurare un semplice processo server conAmazon GameLift Servers.

```
//initSDK
var initSDKOutcome = GameLiftServerAPI.InitSDK();

//processReady
// Set parameters and call ProcessReady
var processParams = new ProcessParameters(
    this.OnGameSession,
    this.OnProcessTerminate,
    this.OnHealthCheck,
    this.OnGameSessionUpdate,
    port,
    // Examples of log and error files written by the game server
    new LogParameters(new List<string>()
        {
            "C:\\game\\logs",
            "C:\\game\\error"
        })
);

var processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);

// Implement callback functions
void OnGameSession(GameSession gameSession)
{
```

```
    // game-specific tasks when starting a new game session, such as loading map
    // When ready to receive players
    var activateGameSessionOutcome = GameLiftServerAPI.ActivateGameSession();
}

void OnProcessTerminate()
{
    // game-specific tasks required to gracefully shut down a game session,
    // such as notifying players, preserving game state data, and other cleanup
    var ProcessEndingOutcome = GameLiftServerAPI.ProcessEnding();
}

bool OnHealthCheck()
{
    bool isHealthy;
    // complete health evaluation within 60 seconds and set health
    return isHealthy;
}
```

Inizia una sessione di gioco

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Una volta completata l'inizializzazione del gioco, puoi iniziare una sessione di gioco.

1. Implementa la funzione di callback. on`StartGameSession` Amazon GameLift Serversrichiama questo metodo per avviare una nuova sessione di gioco sul processo server e ricevere le connessioni dei giocatori.
2. Per attivare una sessione di gioco, chiama. `ActivateGameSession()` Per ulteriori informazioni sull'SDK, consultaSDK del server C# per Amazon GameLift Servers 4.x -- Azioni.

Il seguente esempio di codice illustra come avviare una sessione di gioco con. Amazon GameLift Servers

```
void OnStartGameSession(GameSession gameSession)
{
```

```
    // game-specific tasks when starting a new game session, such as loading map
    ...
    // When ready to receive players
    var activateGameSessionOutcome = GameLiftServerAPI.ActivateGameSession();
}
```

Termina una sessione di gioco

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Notifica Amazon GameLift Servers quando una sessione di gioco sta per terminare. Come best practice, chiudi i processi del server al termine delle sessioni di gioco per riciclare e aggiornare le risorse di hosting.

1. Imposta una funzione denominata on`ProcessTerminate` per ricevere richieste Amazon GameLift Servers e chiamate. `ProcessEnding()`
2. Lo stato del processo cambia in`TERMINATED`.

L'esempio seguente descrive come terminare un processo per una sessione di gioco.

```
var processEndingOutcome = GameLiftServerAPI.ProcessEnding();

if (processReadyOutcome.Success)
   Environment.Exit(0);

// otherwise, exit with error code
Environment.Exit(errorCode);
```

Crea server, costruisci e carica su Amazon GameLift Servers

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Dopo aver integrato il server di gioco conAmazon GameLift Servers, carica i file di build su una flotta in modo che Amazon GameLift Servers possa utilizzarli per l'hosting dei giochi. Per ulteriori informazioni su come caricare il server suAmazon GameLift Servers, consultaImplementa una build di server personalizzata per Amazon GameLift Servers hosting.

Integrazione Amazon GameLift Servers con un progetto client di gioco Unity

> **Note**
>
> Questo argomento fornisce informazioni su una versione precedente del Amazon GameLift Servers plug-in per Unity. La versione 1.0.0 (rilasciata nel 2021) utilizza l'SDK del server per Amazon GameLift Servers 4.x o versioni precedenti. Per la documentazione sull'ultima versione del plug-in, che utilizza il server SDK 5.x e supporta Anywhere, consulta. Amazon GameLift Servers Amazon GameLift Serversplugin per Unity (server SDK 5.x)

Questo argomento ti aiuta a configurare un client di gioco per connetterti alle sessioni di gioco Amazon GameLift Servers ospitate tramite un servizio di backend. Utilizzalo Amazon GameLift Servers APIs per avviare il matchmaking, richiedere il posizionamento della sessione di gioco e altro ancora.

Aggiungi codice al progetto del servizio di backend per consentire la comunicazione con il servizio. Amazon GameLift Servers Un servizio di backend gestisce tutte le comunicazioni del client di gioco con il GameLift servizio. Per ulteriori informazioni sui servizi di backend, consulta.

Un server di backend gestisce le seguenti attività del client di gioco:

- Personalizza l'autenticazione per i tuoi giocatori.
- Richiedere informazioni su sessioni di gioco attive dal servizio Amazon GameLift Servers.
- Crea una nuova sessione di gioco.
- Aggiungi un giocatore a una sessione di gioco esistente.
- Rimuovi un giocatore da una sessione di gioco esistente.

Argomenti

- Aggiungere giocatori alle sessioni di gioco
- Rimuovere un giocatore da una sessione di gioco

Prerequisiti

Prima di configurare la comunicazione del server di gioco con il Amazon GameLift Servers client, completa le seguenti attività:

- Configura un Account AWS
- Installa e configura il plugin
- Integrazione Amazon GameLift Servers con un progetto di server di gioco Unity
- Configurazione di una flotta di hosting con Amazon GameLift Servers

Inizializza un client di gioco

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plugin per la versione 1.0.0 di Unity, che utilizza il server SDK 4.x o precedente.

Aggiungi codice per inizializzare un client di gioco. Esegui questo codice all'avvio, è necessario per altre Amazon GameLift Servers funzioni.

1. Inizializza`AmazonGameLiftClient`. Chiama `AmazonGameLiftClient` con una configurazione client predefinita o una configurazione personalizzata. Per ulteriori informazioni su come configurare un client, vedereConfigura Amazon GameLift Servers su un servizio di backend.
2. Genera un ID giocatore univoco per ogni giocatore per connettersi a una sessione di gioco. Per ulteriori informazioni, consulta Genera giocatore IDs.

   Gli esempi seguenti mostrano come configurare un Amazon GameLift Servers client.

```
public class GameLiftClient
{
    private GameLift gl;
    //A sample way to generate random player IDs.
    bool includeBrackets = false;
```

```
    bool includeDashes = true;
    string playerId = AZ::Uuid::CreateRandom().ToString<string>(includeBrackets,
 includeDashes);


    private Amazon.GameLift.Model.PlayerSession psession = null;
    public AmazonGameLiftClient aglc = null;

    public void CreateGameLiftClient()
    {
        //Access Amazon GameLift Servers service by setting up a configuration.
        //The default configuration specifies a location.
        var config = new AmazonGameLiftConfig();
        config.RegionEndpoint = Amazon.RegionEndpoint.USEast1;

        CredentialProfile profile = null;
        var nscf = new SharedCredentialsFile();
        nscf.TryGetProfile(profileName, out profile);
        AWSCredentials credentials = profile.GetAWSCredentials(null);
        //Initialize Amazon GameLift Servers Client with default client
 configuration.
        aglc = new AmazonGameLiftClient(credentials, config);

    }
}
```

Crea una sessione di gioco su una flotta specifica

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per la versione 1.0.0 di Unity, che utilizza il server SDK 4.x o precedente.

Aggiungere codice per avviare nuove sessioni di gioco nei parchi di istanze distribuite e renderle disponibili ai giocatori. Dopo Amazon GameLift Servers aver creato la nuova sessione di gioco e aver restituito un`GameSession`, puoi aggiungervi giocatori.

- Effettua una richiesta per una nuova sessione di gioco.
  - Se il gioco utilizza flotte, chiama `CreateGameSession()` con un ID flotta o alias, un nome di sessione e il numero massimo di giocatori simultanei per la partita.

- Se il gioco utilizza code, chiama. `StartGameSessionPlacement()`

L'esempio seguente mostra come creare una sessione di gioco.

```
public Amazon.GameLift.Model.GameSession()
{
    var cgsreq = new Amazon.GameLift.Model.CreateGameSessionRequest();
    //A unique identifier for the alias with the fleet to create a game session in.
    cgsreq.AliasId = aliasId;
    //A unique identifier for a player or entity creating the game session
    cgsreq.CreatorId = playerId;
    //The maximum number of players that can be connected simultaneously to the game
 session.
    cgsreq.MaximumPlayerSessionCount = 4;

    //Prompt an available server process to start a game session and retrieves
 connection information for the new game session
    Amazon.GameLift.Model.CreateGameSessionResponse cgsres =
 aglc.CreateGameSession(cgsreq);
    string gsid = cgsres.GameSession != null ? cgsres.GameSession.GameSessionId : "N/
A";
    Debug.Log((int)cgsres.HttpStatusCode + " GAME SESSION CREATED: " + gsid);
    return cgsres.GameSession;
}
```

Aggiungere giocatori alle sessioni di gioco

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Dopo Amazon GameLift Servers aver creato la nuova sessione di gioco e aver restituito un `GameSession` oggetto, puoi aggiungervi giocatori.

1. Riserva uno slot per giocatori in una sessione di gioco creando una nuova sessione di gioco. Usa `CreatePlayerSession` o `CreatePlayerSessions` con l'ID della sessione di gioco e un ID univoco per ogni giocatore.

2. Connect alla sessione di gioco. Recupera l'`PlayerSession`oggetto per ottenere le informazioni di connessione della sessione di gioco. Puoi usare queste informazioni per stabilire una connessione diretta al processo del server:

   a. Utilizzate la porta specificata e il nome DNS o l'indirizzo IP del processo server.

   b. Usa il nome DNS e la porta delle tue flotte. Il nome e la porta DNS sono obbligatori se le tue flotte hanno abilitato la generazione di certificati TLS.

   c. Fai riferimento all'ID della sessione del giocatore. L'ID della sessione del giocatore è necessario se il server di gioco convalida le connessioni dei giocatori in entrata.

Gli esempi seguenti mostrano come riservare un posto giocatore in una sessione di gioco.

```
public Amazon.GameLift.Model.PlayerSession
 CreatePlayerSession(Amazon.GameLift.Model.GameSession gsession)
{
    var cpsreq = new Amazon.GameLift.Model.CreatePlayerSessionRequest();
    cpsreq.GameSessionId = gsession.GameSessionId;
    //Specify game session ID.
    cpsreq.PlayerId = playerId;
    //Specify player ID.
    Amazon.GameLift.Model.CreatePlayerSessionResponse cpsres =
 aglc.CreatePlayerSession(cpsreq);
    string psid = cpsres.PlayerSession != null ? cpsres.PlayerSession.PlayerSessionId :
 "N/A";
    return cpsres.PlayerSession;
}
```

Il codice seguente illustra come connettere un giocatore alla sessione di gioco.

```
public bool ConnectPlayer(int playerIdx, string playerSessionId)
{
    //Call ConnectPlayer with player ID and player session ID.
    return server.ConnectPlayer(playerIdx, playerSessionId);
}
```

Rimuovere un giocatore da una sessione di gioco

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per la versione 1.0.0 di Unity, che utilizza il server SDK 4.x o precedente.

Puoi rimuovere i giocatori dalla sessione di gioco quando escono dal gioco.

1. Notifica al Amazon GameLift Servers servizio che un giocatore si è disconnesso dal processo del server. Chiama `RemovePlayerSession` con l'ID di sessione del giocatore.
2. Verifica che venga `RemovePlayerSession` restituito`Success`. Quindi, Amazon GameLift Servers cambia lo slot del giocatore rendendolo disponibile, che Amazon GameLift Servers può assegnare a un nuovo giocatore.

L'esempio seguente illustra come rimuovere una sessione di gioco.

```
public void DisconnectPlayer(int playerIdx)
{
    //Receive the player session ID.
    string playerSessionId = playerSessions[playerIdx];
    var outcome = GameLiftServerAPI.RemovePlayerSession(playerSessionId);
    if (outcome.Success)
    {
        Debug.Log (":) PLAYER SESSION REMOVED");
    }
    else
    {
        Debug.Log(":(PLAYER SESSION REMOVE FAILED. RemovePlayerSession()
        returned " + outcome.Error.ToString());
    }
}
```

Installa e configura il plugin

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

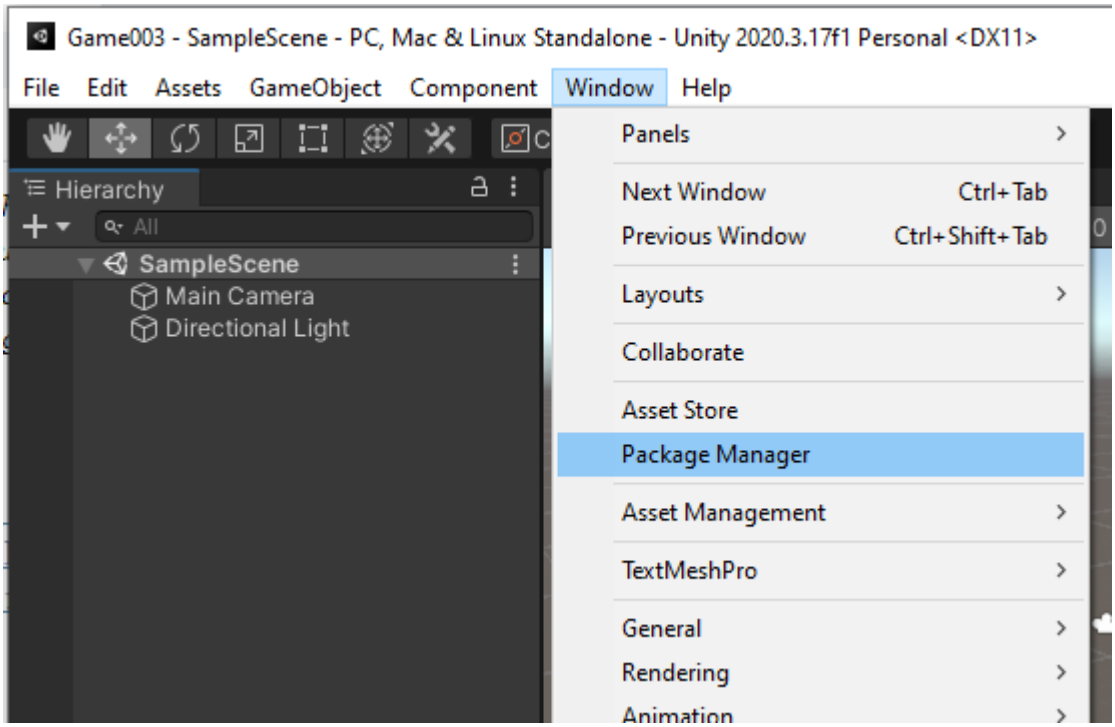
Questa sezione descrive come scaricare, installare e configurare il Amazon GameLift Servers plug-in per Unity, versione 1.0.0.

Prerequisiti

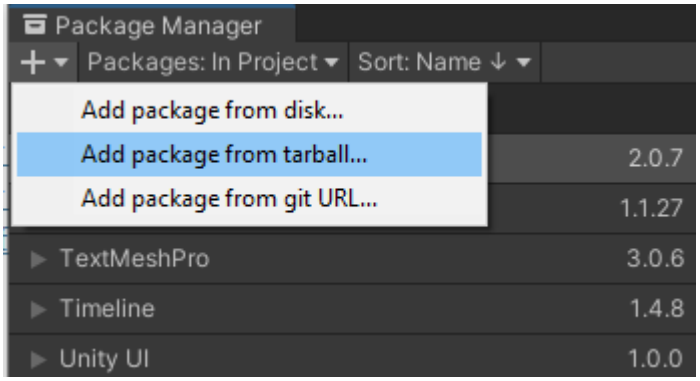
- Unity per Windows 2019.4 LTS, Windows 2020.3 LTS o Unity per macOS
- Versione attuale di Java
- Versione attuale di.NET 4.x

Per scaricare e installare il plugin per Unity

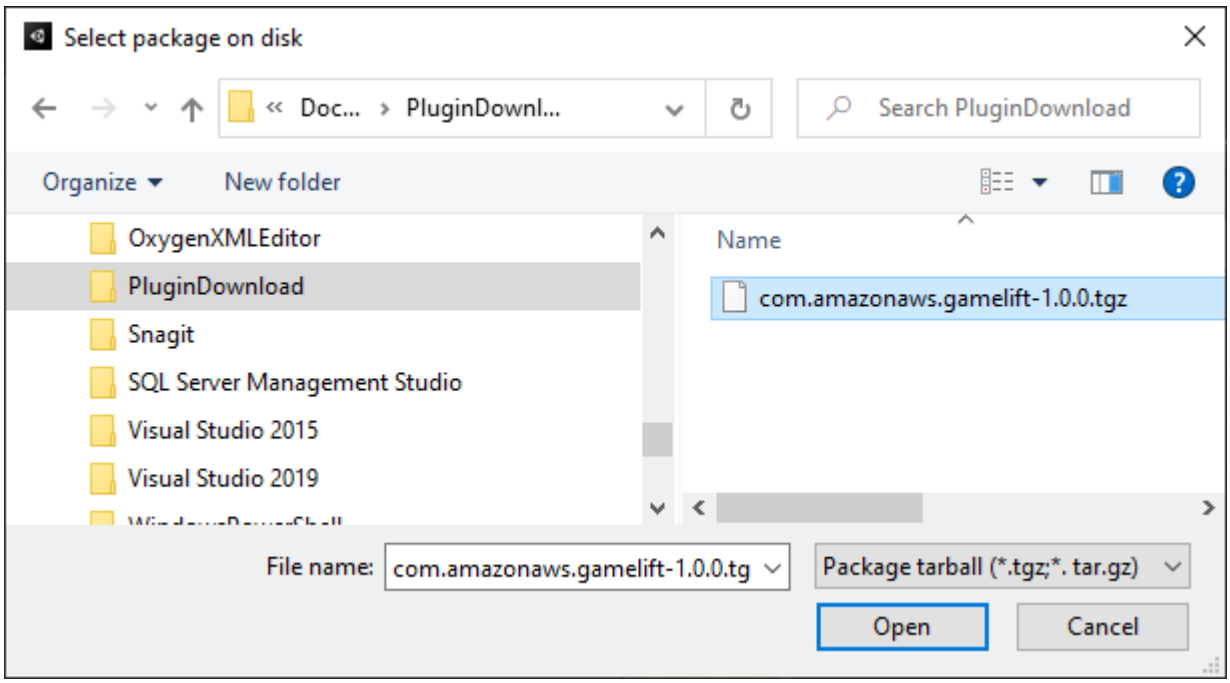
1. Scarica il Amazon GameLift Servers plugin per Unity. Puoi trovare l'ultima versione nella pagina del repository del Amazon GameLift Servers plugin per Unity. Nella versione più recente, scegli Risorse, quindi scarica il `com.amazonaws.gamelift-`*`version`*`.tgz` file.
2. Avvia Unity e scegli un progetto.
3. Nella barra di navigazione in alto, sotto Finestra scegli Package Manager:



4. Nella scheda Package Manager scegli +, quindi scegli Aggiungi pacchetto da tarball... :



5. Nella finestra Seleziona pacchetti su disco, accedi alla `com.amazonaws.gamelift` cartella, scegli il file`com.amazonaws.gamelift-`*`version`*`.tgz` , quindi scegli Apri:



6. Dopo che Unity ha caricato il plug-in, Amazon GameLift Serversappare come una nuova voce nel menu Unity. Potrebbero essere necessari alcuni minuti per installare e ricompilare gli script. La scheda Impostazioni del Amazon GameLift Servers plugin si apre automaticamente.

7. Nel riquadro SDK, scegli Use.NET 4.x.

   Una volta configurato, lo stato cambia da Non configurato a Configurato.

Prova il gioco localmente

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plugin per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Usa Amazon GameLift Servers Local per eseguirlo Amazon GameLift Servers sul tuo dispositivo locale. Puoi usare Amazon GameLift Servers Local per verificare le modifiche al codice in pochi secondi, senza una connessione di rete.

Configura i test locali

1. Nella finestra del plugin per Unity, scegli la scheda Test.
2. Nel riquadro Test, scegli Download Amazon GameLift Servers locale. Il plugin per Unity apre una finestra del browser e scarica il `GameLift_06_03_2021.zip` file nella cartella dei download.

   Il download include l'SDK del server C#, i file di sorgente.NET e un component.NET compatibile con Unity.
3. Decomprimere il file `GameLift_06_03_2021.zip` scaricato.
4. Nella finestra Impostazioni del Amazon GameLift Servers plug-in, scegliete Percorso Amazon GameLift Servers locale, accedete alla cartella decompressa, scegliete il file**`GameLiftLocal.jar`**, quindi scegliete Apri.

   Una volta configurato, lo stato del test locale cambia da Non configurato a Configurato.
5. Verifica lo stato del JRE. Se lo stato è Non configurato, scegli Scarica JRE e installa la versione Java consigliata.

   Dopo aver installato e configurato l'ambiente Java, lo stato cambia in Configurato.

Esegui il gioco locale

1. Nella scheda del plugin per Unity, scegli la scheda Test.
2. Nel riquadro Test, scegli Open Local Test UI.
3. Nella finestra Local Testing, specifica un percorso eseguibile del server. Seleziona... per selezionare il percorso e il nome eseguibile dell'applicazione server.

4. Nella finestra Local Testing, specifica una porta GL Local.
5. Scegli Deploy & Run per distribuire ed eseguire il server.
6. Per arrestare il server di gioco, scegli Stop o chiudi le finestre del server di gioco.

Implementa uno scenario

> **Note**
>
> Questo argomento fa riferimento al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Uno scenario utilizza un AWS CloudFormation modello per creare le risorse necessarie per implementare una soluzione di hosting cloud per il gioco. Questa sezione descrive gli scenari Amazon GameLift Servers forniti e come utilizzarli.

Prerequisiti

Per implementare lo scenario, è necessario un ruolo IAM per il Amazon GameLift Servers servizio. Per informazioni su come creare un ruolo perAmazon GameLift Servers, consultaConfigura un Account AWS.

Ogni scenario richiede le autorizzazioni per le seguenti risorse:

- Amazon GameLift Servers
- Amazon S3
- AWS CloudFormation
- API Gateway
- AWS Lambda
- AWS WAFV2
- Amazon Cognito

Scenari

> **Note**
>
> Questo argomento fa riferimento al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Il Amazon GameLift Servers plug-in per Unity include i seguenti scenari:

Solo autenticazione

Questo scenario crea un servizio di backend di gioco che esegue l'autenticazione dei giocatori senza la funzionalità del server di gioco. Il modello crea le seguenti risorse nel tuo account:

- Un pool di utenti Amazon Cognito per archiviare le informazioni di autenticazione dei giocatori.
- Un AWS Lambda gestore basato su endpoint REST di Amazon API Gateway che avvia giochi e visualizza le informazioni sulla connessione del gioco.

Flotta per una singola regione

Questo scenario crea un servizio di backend di gioco con un'unica Amazon GameLift Servers flotta. Crea le seguenti risorse:

- Un pool di utenti Amazon Cognito per consentire a un giocatore di autenticarsi e iniziare una partita.
- Un AWS Lambda gestore per la ricerca di una sessione di gioco esistente con uno slot per giocatori aperto sulla flotta. Se non riesce a trovare uno slot libero, crea una nuova sessione di gioco.

Flotta multiregionale con coda e matchmaker personalizzato

Questo scenario crea partite utilizzando Amazon GameLift Servers code e un matchmaker personalizzato per raggruppare i giocatori più anziani nel pool d'attesa. Crea le seguenti risorse:

- Un argomento di Amazon Simple Notification Service su cui vengono Amazon GameLift Servers pubblicati messaggi. Per ulteriori informazioni sugli argomenti e sulle notifiche SNS, consulta. Imposta la notifica degli eventi per il posizionamento della sessione di gioco

- Una funzione Lambda richiamata dal messaggio che comunica i dettagli del posizionamento e della connessione al gioco.
- Una tabella Amazon DynamoDB per memorizzare i dettagli del posizionamento e della connessione al gioco. `GetGameConnection`le chiamate vengono lette da questa tabella e restituiscono le informazioni di connessione al client di gioco.

Individua le flotte con una coda e un matchmaker personalizzato

Questo scenario crea le partite utilizzando Amazon GameLift Servers code e un matchmaker personalizzato e configura tre flotte. Crea le seguenti risorse:

- Due flotte Spot che contengono diversi tipi di istanze per garantire la durabilità in caso di indisponibilità di Spot.
- Una flotta On-Demand che funge da backup per le altre flotte Spot. Per ulteriori informazioni sulla progettazione delle flotte, consulta. Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite
- Una Amazon GameLift Servers coda per mantenere alta la disponibilità dei server e ridurre i costi. Per ulteriori informazioni e procedure consigliate sulle code, vedere. Personalizza una coda di sessioni di gioco

FlexMatch

Questo scenario utilizza FlexMatch, un servizio di matchmaking gestito, per abbinare i giocatori. Per ulteriori informazioni suFlexMatch, vedi What is Amazon GameLift ServersFlexMatch. Questo scenario crea le seguenti risorse:

- Una funzione Lambda per creare un ticket di matchmaking dopo aver ricevuto le richieste. `StartGame`
- Una funzione Lambda separata per ascoltare gli eventi delle FlexMatch partite.

Per evitare costi inutili Account AWS, rimuovi le risorse create da ogni scenario dopo averle utilizzate. Elimina lo AWS CloudFormation stack corrispondente.

Aggiorna le credenziali AWS

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Il Amazon GameLift Servers plug-in per Unity richiede credenziali di sicurezza per distribuire uno scenario. È possibile creare nuove credenziali o utilizzare credenziali esistenti.

Per ulteriori informazioni sulla configurazione delle credenziali, consulta Comprendere e ottenere le credenziali. AWS

Per aggiornare le credenziali AWS

1. In Unity, nel plugin per la scheda Unity, scegli la scheda Deploy.
2. Nel riquadro Distribuisci, scegli AWS Credenziali.
3. Puoi creare nuove AWS credenziali o scegliere credenziali esistenti.
   - Per creare credenziali, scegli Crea nuovo profilo di credenziali, quindi specifica il nuovo nome del profilo, l'ID della chiave di AWS accesso, la chiave AWS segreta e. Regione AWS
   - Per scegliere le credenziali esistenti, scegli il profilo Scegli le credenziali esistenti, quindi seleziona un nome di profilo e. Regione AWS
4. Nella finestra Aggiorna AWS credenziali, scegli Aggiorna profilo credenziali.

Aggiorna il bootstrap dell'account

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

La posizione di bootstrap è un bucket Amazon S3 utilizzato durante la distribuzione. Viene utilizzato per archiviare le risorse dei server di gioco e altre dipendenze. Il bucket Regione AWS scelto deve corrispondere alla stessa regione che utilizzerai per la distribuzione dello scenario.

Per ulteriori informazioni sui bucket Amazon S3, consulta Creazione, configurazione e utilizzo dei bucket Amazon Simple Storage Service.

Per aggiornare la posizione di avvio dell'account

1. In Unity, nel plugin per la scheda Unity, scegli la scheda Deploy.
2. Nel riquadro Deploy, scegli Update Account Bootstrap.
3. Nella finestra Account Bootstrapping, scegli un bucket Amazon S3 esistente o creane uno nuovo:
   - Per scegliere un bucket esistente, scegli Scegli un bucket Amazon S3 esistente e Aggiorna per salvare la selezione.
   - Scegli Crea nuovo bucket Amazon S3 per creare un nuovo bucket Amazon Simple Storage Service, quindi scegli una policy. La policy specifica quando scadrà il bucket Amazon S3. Scegli Crea per creare il bucket.

Implementa uno scenario di gioco

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

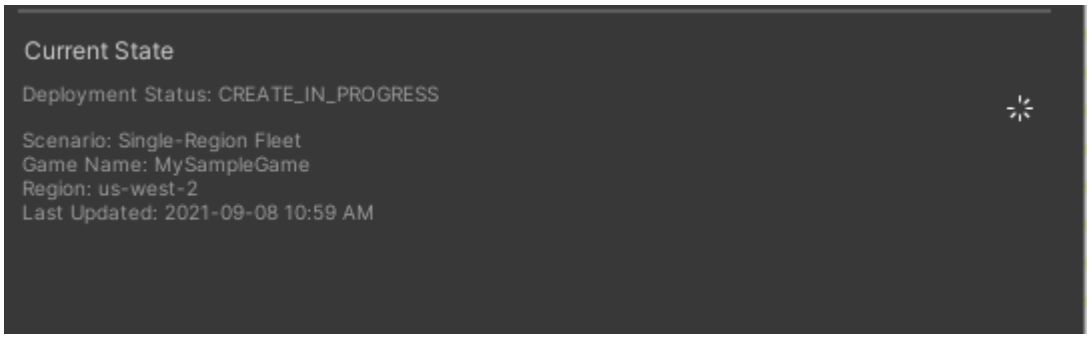
Puoi usare uno scenario per testare il tuo gioco. Amazon GameLift Servers Ogni scenario utilizza un AWS CloudFormation modello per creare uno stack con le risorse richieste. La maggior parte degli scenari richiede un file eseguibile sul server di gioco e un percorso di compilazione. Quando distribuisci lo scenario, Amazon GameLift Servers copia le risorse di gioco nella posizione di bootstrap come parte della distribuzione.

È necessario configurare AWS le credenziali e un AWS account bootstrap per implementare uno scenario.

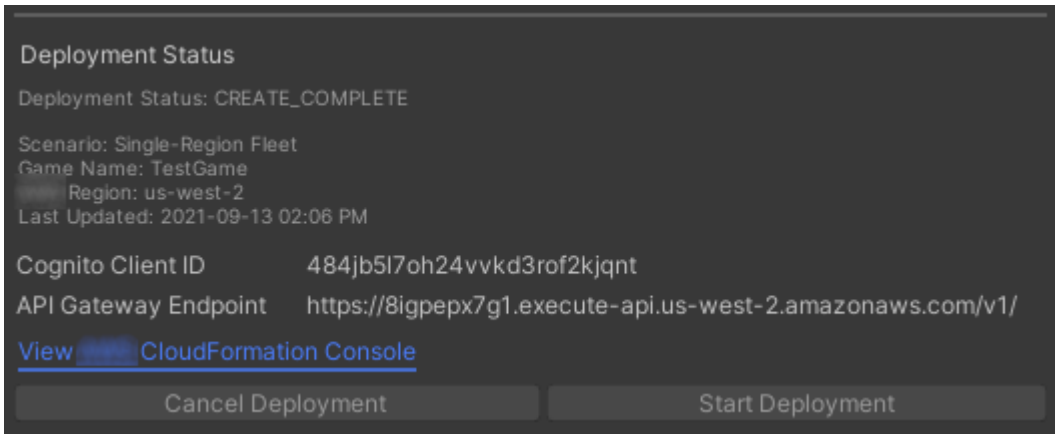
Per implementare uno scenario

1. In Unity, nel plugin per la scheda Unity, scegli la scheda Deploy.
2. Nel riquadro Deploy, scegli Open Deployment UI.
3. Nella finestra Distribuzione, scegli uno scenario.

4. Inserisci un nome per il gioco. Deve essere univoco. Il nome del gioco fa parte del nome AWS CloudFormation dello stack quando distribuisci lo scenario.
5. Scegli il percorso della cartella di creazione del server di gioco. Il percorso della cartella di compilazione punta alla cartella contenente l'eseguibile e le dipendenze del server.
6. Scegli il percorso del file Game Server Build .exe. Il percorso del file eseguibile di build punta all'eseguibile del server di gioco.
7. Scegli Avvia distribuzione per iniziare a distribuire uno scenario. Puoi seguire lo stato dell'aggiornamento nella finestra Distribuzione in Stato corrente. La distribuzione degli scenari può richiedere diversi minuti.



8. Quando lo scenario completa la distribuzione, lo stato attuale si aggiorna per includere l'ID client Cognito e l'endpoint API Gateway che puoi copiare e incollare nel gioco.



9. Per aggiornare le impostazioni di gioco, nel menu Unity, scegli Vai alle impostazioni di connessione del client. Viene visualizzata una scheda Inspector sul lato destro della schermata Unity.
10. Deseleziona la modalità di test locale.
11. Inserite l'API Gateway Endpoint e l'ID client Coginito. Scegli lo stesso che Regione AWS hai usato per l'implementazione dello scenario. Puoi quindi ricostruire ed eseguire il client di gioco utilizzando le risorse dello scenario distribuite.

Eliminazione delle risorse create dallo scenario

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Per eliminare le risorse create per lo scenario, eliminate lo stack corrispondente. AWS CloudFormation

Per eliminare le risorse create dallo scenario

1. Nella finestra del Amazon GameLift Servers plug-in per Unity Deployment, seleziona Visualizza AWS CloudFormation console per aprire la AWS CloudFormation console.
2. Nella AWS CloudFormation console, scegli Stacks, quindi scegli lo stack che include il nome del gioco specificato durante la distribuzione.
3. Scegli Elimina per eliminare lo stack. Potrebbero essere necessari alcuni minuti per eliminare uno stack. Dopo aver AWS CloudFormation eliminato lo stack utilizzato dallo scenario, il suo stato diventa. `ROLLBACK_COMPLETE`

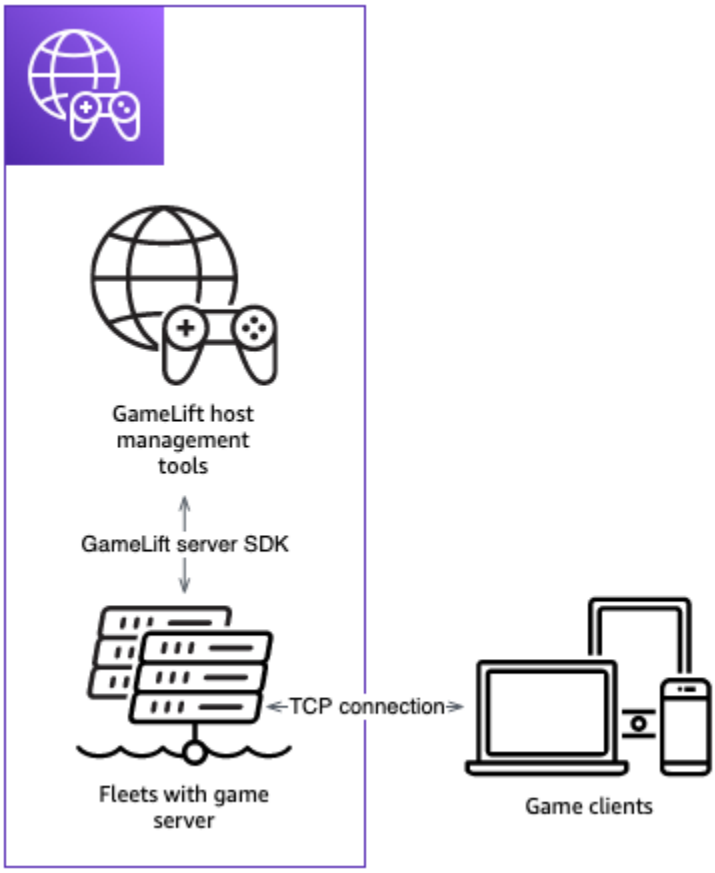
Integra i giochi con Unity Amazon GameLift Servers

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Integra il tuo gioco Unity con Amazon GameLift Servers completando le seguenti attività:

- Integrazione Amazon GameLift Servers con un progetto di server di gioco Unity
- Integrazione Amazon GameLift Servers con un progetto client di gioco Unity

Il diagramma seguente mostra un esempio di flusso di integrazione di un gioco. Nel diagramma, viene schierata una flotta con il server di gioco. Amazon GameLift Servers Il client di gioco comunica con il server di gioco, che comunica con. Amazon GameLift Servers



Importa ed esegui un gioco di esempio

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Il Amazon GameLift Servers plugin per Unity include un gioco di esempio che puoi usare per esplorare le basi dell'integrazione con il tuo gioco. Amazon GameLift Servers In questa sezione, crei il client e il server di gioco e poi esegui i test localmente utilizzando Amazon GameLift Servers Local.

Prerequisiti

- Configura un Account AWS
- Installa e configura il plugin

Crea ed esegui il server di gioco di esempio

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per la versione 1.0.0 di Unity, che utilizza il server SDK 4.x o precedente.

Configura i file del server di gioco del gioco di esempio.

1. In Unity, nel menu, scegli Amazon GameLift Servers, quindi scegli Importa gioco di esempio.
2. Nella finestra Importa gioco di esempio, scegli Importa per importare il gioco, le sue risorse e le sue dipendenze.
3. Costruisci il server di gioco. In Unity, nel menu, scegli Amazon GameLift Servers, quindi scegli Applica impostazioni build Windows Sample Server o Applica impostazioni build server di esempio macOS. Dopo aver configurato le impostazioni del server di gioco, Unity ricompila le risorse.
4. In Unity, nel menu, scegli File, quindi scegli Crea. Scegli Server Build, scegli Build, quindi scegli una cartella di build specifica per i file del server.

   Unity crea il server di gioco di esempio, inserendo l'eseguibile e le risorse richieste nella cartella di build specificata.

Crea ed esegui il client di gioco di esempio

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per la versione 1.0.0 di Unity, che utilizza il server SDK 4.x o precedente.

Configura i file client di gioco del gioco di esempio.

1. In Unity, nel menu, scegli Amazon GameLift Servers, quindi scegli Applica impostazioni di build del client di esempio di Windows o Applica impostazioni di build del client di esempio macOS. Dopo aver configurato le impostazioni del client di gioco, Unity ricompilerà le risorse.

2. In Unity, nel menu, seleziona Vai alle impostazioni del client. Verrà visualizzata una scheda Inspector sul lato destro della schermata Unity. Nella scheda Impostazioni Amazon GameLift Servers client, seleziona Local Testing Mode.
3. Crea il client di gioco. In Unity, nel menu, scegli File. Conferma che Server Build non sia selezionato, scegli Build, quindi seleziona una cartella di build specifica per i file client.

   Unity crea il client di gioco di esempio, inserendo l'eseguibile e le risorse richieste nella cartella di build del client specificata.
4. Non hai creato il server e il client di gioco. Nei passaggi successivi, esegui il gioco e vedi come interagisce conAmazon GameLift Servers.

Prova il gioco di esempio a livello locale

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Esegui il gioco di esempio che hai importato utilizzando Local. Amazon GameLift Servers

1. Avvia il server di gioco. In Unity, nel plugin per la scheda Unity, scegli la scheda Deploy.
2. Nel riquadro Test, scegli Open Local Test UI.
3. Nella finestra Local Testing, specifica il percorso del file.exe di Game Server. Il percorso deve includere il nome dell'eseguibile. Ad esempio, `C:/MyGame/GameServer/MyGameServer.exe`.
4. Scegli Deploy and Run. Il plug-in per Unity avvia il server di gioco e apre una finestra di registro Amazon GameLift Servers locale. La finestra contiene messaggi di registro, inclusi i messaggi inviati tra il server di gioco e Amazon GameLift Servers Local.
5. Avvia il client di gioco. Trova la posizione di costruzione con il client di gioco di esempio e scegli il file eseguibile.
6. Nel gioco Amazon GameLift Servers di esempio, fornisci un'e-mail e una password, quindi scegli Accedi. L'email e la password non vengono convalidate o utilizzate.
7. Nel gioco di Amazon GameLift Servers esempio, scegli Inizia. Il client di gioco cerca una sessione di gioco. Se non riesce a trovare una sessione, ne crea una. Il client di gioco avvia quindi la sessione di gioco. Puoi vedere l'attività di gioco nei registri.

Esempi di log del server di gioco

```
...
2021-09-15T19:55:3495 PID:20728 Log :) GAMELIFT AWAKE
2021-09-15T19:55:3512 PID:20728 Log :) I AM SERVER
2021-09-15T19:55:3514 PID:20728 Log :) GAMELIFT StartServer at port 33430.
2021-09-15T19:55:3514 PID:20728 Log :) SDK VERSION: 4.0.2
2021-09-15T19:55:3556 PID:20728 Log :) SERVER IS IN A GAMELIFT FLEET
2021-09-15T19:55:3577 PID:20728 Log :) PROCESSREADY SUCCESS.
2021-09-15T19:55:3577 PID:20728 Log :) GAMELIFT HEALTH CHECK REQUESTED (HEALTHY)
...
2021-09-15T19:55:3634 PID:20728 Log :) GAMELOGIC AWAKE
2021-09-15T19:55:3635 PID:20728 Log :) GAMELOGIC START
2021-09-15T19:55:3636 PID:20728 Log :) LISTENING ON PORT 33430
2021-09-15T19:55:3636 PID:20728 Log SERVER: Frame: 0 HELLO WORLD!
...
2021-09-15T19:56:2464 PID:20728 Log :) GAMELIFT SESSION REQUESTED
2021-09-15T19:56:2468 PID:20728 Log :) GAME SESSION ACTIVATED
2021-09-15T19:56:3578 PID:20728 Log :) GAMELIFT HEALTH CHECK REQUESTED (HEALTHY)
2021-09-15T19:57:3584 PID:20728 Log :) GAMELIFT HEALTH CHECK REQUESTED (HEALTHY)
2021-09-15T19:58:0334 PID:20728 Log SERVER: Frame: 8695 Connection accepted: playerIdx
 0 joined
2021-09-15T19:58:0335 PID:20728 Log SERVER: Frame: 8696 Connection accepted: playerIdx
 1 joined
2021-09-15T19:58:0338 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from playerIdx 0 Msg:
 CONNECT: server IP localhost
2021-09-15T19:58:0338 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from player 0:CONNECT:
 server IP localhost
2021-09-15T19:58:0339 PID:20728 Log SERVER: Frame: 8697 CONNECT: player index 0
2021-09-15T19:58:0339 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from playerIdx 1 Msg:
 CONNECT: server IP localhost
2021-09-15T19:58:0339 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from player 1:CONNECT:
 server IP localhost
2021-09-15T19:58:0339 PID:20728 Log SERVER: Frame: 8697 CONNECT: player index 1
```

Esempi di log Amazon GameLift Servers locali

```
12:55:26,000  INFO || - [SocketIOServer] main - Session store / pubsub factory used:
 MemoryStoreFactory (local session store only)
12:55:28,092  WARN || - [ServerBootstrap] main - Unknown channel option 'SO_LINGER' for
 channel '[id: 0xe23d0a14]'
12:55:28,101  INFO || - [SocketIOServer] nioEventLoopGroup-2-1 - SocketIO server
 started at port: 5757
```

```
12:55:28,101  INFO || - [SDKConnection] main - GameLift SDK server (communicates with
 your game server) has started on http://localhost:5757
12:55:28,120  INFO || - [SdkWebSocketServer] WebSocketSelector-20 - WebSocket Server
 started on address localhost/127.0.0.1:5759
12:55:28,166  INFO || - [StandAloneServer] main - GameLift Client server (listens for
 GameLift client APIs) has started on http://localhost:8080
12:55:28,179  INFO || - [StandAloneServer] main - GameLift server sdk http listener has
 started on http://localhost:5758
12:55:35,453  INFO || - [SdkWebSocketServer] WebSocketWorker-12 - onOpen
 socket: /?pID=20728&sdkVersion=4.0.2&sdkLanguage=CSharp and handshake /?
pID=20728&sdkVersion=4.0.2&sdkLanguage=CSharp
12:55:35,551  INFO || - [HostProcessManager] WebSocketWorker-12 - client connected with
 pID 20728
12:55:35,718  INFO || - [GameLiftSdkHttpHandler] GameLiftSdkHttpHandler-thread-0 -
 GameLift API to use: ProcessReady for pId 20728
12:55:35,718  INFO || - [ProcessReadyHandler] GameLiftSdkHttpHandler-thread-0 -
 Received API call for processReady from 20728
12:55:35,738  INFO || - [ProcessReadyHandler] GameLiftSdkHttpHandler-thread-0 -
 onProcessReady data: port: 33430
 12:55:35,739  INFO || - [HostProcessManager] GameLiftSdkHttpHandler-thread-0 -
 Registered new process with pId 20728
12:55:35,789  INFO || - [GameLiftSdkHttpHandler] GameLiftSdkHttpHandler-thread-0 -
 GameLift API to use: ReportHealth for pId 20728
12:55:35,790  INFO || - [ReportHealthHandler] GameLiftSdkHttpHandler-thread-0 -
 Received API call for ReportHealth from 20728
12:55:35,794  INFO || - [ReportHealthHandler] GameLiftSdkHttpHandler-thread-0 -
 ReportHealth data: healthStatus: true
 12:56:24,098  INFO || - [GameLiftHttpHandler] Thread-12 - API to use:
 GameLift.DescribeGameSessions
12:56:24,119  INFO || - [DescribeGameSessionsDispatcher] Thread-12 - Received API call
 to describe game sessions with input: {"FleetId":"fleet-123"}
12:56:24,241  INFO || - [GameLiftHttpHandler] Thread-12 - API to use:
 GameLift.CreateGameSession
12:56:24,242  INFO || - [CreateGameSessionDispatcher] Thread-12 - Received API call to
 create game session with input: {"FleetId":"fleet-123","MaximumPlayerSessionCount":4}
12:56:24,265  INFO || - [HostProcessManager] Thread-12 - Reserved process:
 20728 for gameSession: arn:aws:gamelift:local::gamesession/fleet-123/
gsess-59f6cc44-4361-42f5-95b5-fdb5825c0f3d
12:56:24,266  INFO || - [WebSocketInvoker] Thread-12 - StartGameSessionRequest:
 gameSessionId=arn:aws:gamelift:local::gamesession/fleet-123/
gsess-59f6cc44-4361-42f5-95b5-fdb5825c0f3d, fleetId=fleet-123, gameSessionName=null,
 maxPlayers=4, properties=[], ipAddress=127.0.0.1, port=33430, gameSessionData?=false,
 matchmakerData?=false, dnsName=localhost
```

```
12:56:24,564  INFO || - [CreateGameSessionDispatcher] Thread-12 - GameSession with
 id: arn:aws:gamelift:local::gamesession/fleet-123/gsess-59f6cc44-4361-42f5-95b5-
fdb5825c0f3d created
12:56:24,585  INFO || - [GameLiftHttpHandler] Thread-12 - API to use:
 GameLift.DescribeGameSessions
12:56:24,585  INFO || - [DescribeGameSessionsDispatcher] Thread-12 - Received API call
 to describe game sessions with input: {"FleetId":"fleet-123"}
12:56:24,660  INFO || - [GameLiftSdkHttpHandler] GameLiftSdkHttpHandler-thread-0 -
 GameLift API to use: GameSessionActivate for pId 20728
12:56:24,661  INFO || - [GameSessionActivateHandler] GameLiftSdkHttpHandler-thread-0 -
 Received API call for GameSessionActivate from 20728
12:56:24,678  INFO || - [GameSessionActivateHandler] GameLiftSdkHttpHandler-thread-0
 - GameSessionActivate data: gameSessionId: "arn:aws:gamelift:local::gamesession/
fleet-123/gsess-59f6cc44-4361-42f5-95b5-fdb5825c0f3d"
```

Arresta il processo del server

> **Note**
>
> Questo argomento si riferisce al Amazon GameLift Servers plug-in per Unity versione 1.0.0, che utilizza il server SDK 4.x o precedente.

Dopo aver finito con il gioco di esempio, spegni il server in Unity.

1. Nel client di gioco, scegli Esci o chiudi la finestra per interrompere il client di gioco.
2. In Unity, nella finestra Local Testing, scegli Stop o chiudi le finestre del server di gioco per arrestare il server.

Roadmap di sviluppo per l'hosting con managed Amazon GameLift Servers EC2

Questa tabella di marcia ti guida nello sviluppo di una soluzione di EC2 hosting Amazon GameLift Servers gestito per il tuo gioco multiplayer. Amazon GameLift Serversoffre diverse opzioni di hosting di giochi; per ulteriori informazioni su queste opzioni, consultaAmazon GameLift Serverssoluzioni.

Con l'hosting Amazon GameLift Servers gestito, il server di gioco è ospitato su risorse informatiche virtuali Cloud AWS basate su risorse di calcolo virtuali che Amazon GameLift Servers possiedono e operano in base alla configurazione dell'utente. Ottieni la sicurezza, l'affidabilità e la disponibilità

globale delle istanze Amazon Elastic Compute Cloud (Amazon EC2) ulteriormente ottimizzate per l'uso con l'hosting di giochi multiplayer. Amazon GameLift Serverssemplifica la gestione dell'hosting con strumenti come l'implementazione automatica dei server, la gestione del ciclo di vita e l'auto-scaling della capacità.

Una soluzione Amazon GameLift Servers gestita è composta dai seguenti componenti:

- Una o più flotte Amazon GameLift Servers gestite, che utilizzano istanze Amazon Elastic Compute Cloud EC2 (Amazon) ottimizzate per l'hosting di giochi multiplayer.
- Una versione di server di gioco, integrata con l'SDK del server perAmazon GameLift Servers, da distribuire su tutte le flotte.
- Un client di gioco e un servizio di backend, integrato con l' AWS SDK, per interagire con il Amazon GameLift Servers servizio e richiedere sessioni di gioco.
- Una Amazon GameLift Servers coda per effettuare nuove sessioni di gioco con i server di gioco disponibili su tutte le flotte.
- (Facoltativo) Un FlexMatch matchmaker per creare partite multigiocatore e impostare sessioni di gioco per esse.

Questa tabella di marcia presenta un percorso semplificato per avviare e far funzionare con successo il gioco multiplayer con l'hosting gestito. Amazon GameLift Servers EC2 Dopo aver installato i componenti necessari, puoi continuare a sviluppare giochi e personalizzare la tua soluzione di hosting. Man mano che ti avvicini al lancio, consulta questi articoli Preparazione del gioco per il lancio con Amazon GameLift Servers hosting per aiutarti a preparare la tua soluzione di hosting per l'utilizzo a livello di produzione.

> **Inizia subito con il Amazon GameLift Servers plugin per Unreal Engine e Unity**
>
> Per una distribuzione più rapida, prova il Amazon GameLift Serversplug-in per Unreal Engine e Unity. Fornisce flussi di lavoro guidati per l'interfaccia utente per implementare rapidamente il server di gioco con una configurazione minima, in modo da poter provare i componenti del gioco in azione. Quindi puoi basarti su queste basi per creare una soluzione di hosting personalizzata per il tuo gioco. Per ulteriori dettagli, consulta Esplora con Amazon GameLift Servers plugin.

Passaggio 1: prepara il server di gioco su cui lavorare Amazon GameLift Servers

Aggiungi funzionalità al tuo server di gioco in modo che possa comunicare con il Amazon GameLift Servers servizio quando viene distribuito per l'hosting.

- Scarica l'SDK del server Amazon GameLift Servers (versione 5.x) per il tuo progetto di gioco. L'SDK del server è disponibile in C++, C# e Go. Scarica un SDK per Amazon GameLift Servers server.
- Modifica il codice del server di gioco per aggiungere la funzionalità SDK del server. Per le linee guida, consulta Integra giochi con server di gioco personalizzati. Come minimo, procedi come segue:
  - Aggiungi codice per inizializzare l'Amazon GameLift ServersSDK e stabilire una WebSocket connessione con il Amazon GameLift Servers servizio. Usa l'azione SDK del server. `InitSdk()`
  - Aggiungi codice da segnalare al Amazon GameLift Servers servizio quando il processo del server è pronto per ospitare sessioni di gioco. Usa l'azione `ProcessReady()` SDK del server.
  - Implementa le funzioni di callback richieste `OnProcessTerminate()` e. `OnStartGameSession()` Con queste funzioni, i processi del server di gioco possono mantenere una connessione con il Amazon GameLift Servers servizio, avviare una sessione di gioco quando richiesto e rispondere alla richiesta di Amazon GameLift Servers terminare il processo del server di gioco.
  - Aggiungi codice per segnalare al Amazon GameLift Servers servizio quando il processo del server sta terminando una sessione di gioco. Usa l'azione `ProcessEnding()` SDK del server.
- Package della build del tuo server di gioco. Crea uno script di installazione con i tuoi file di build, le dipendenze e il software di supporto. Consultare Creare un pacchetto dei file della build di gioco. Ti consigliamo di utilizzare un bucket Amazon Simple Storage Service (Amazon S3) per archiviare le versioni della build del gioco.
- Testa la tua integrazione con i server di gioco. Per questa operazione, consigliamo di configurare una flotta Amazon GameLift Servers Anywhere per una workstation locale, come descritto inConfigura i test locali con Amazon GameLift Servers Anywhere. Per questo passaggio, installa manualmente la build del server di gioco sul dispositivo di prova e avvia un processo server. Utilizza la AWS CLI per richiedere una nuova sessione di gioco e verifica che il Amazon GameLift Servers servizio richieda correttamente al processo del server di avviare una sessione di gioco.

Passaggio 2: prepara il client di gioco per partecipare alle sessioni di gioco ospitate

Crea un modo per consentire al client di gioco di richiedere di partecipare a una sessione di gioco, ottenere informazioni di connessione e quindi connettersi direttamente a una sessione di gioco ospitata. L'approccio più comune consiste nel configurare la funzionalità del servizio di backend che funga da intermediario tra il client di gioco e il servizio. Amazon GameLift Servers Questo approccio protegge le tue risorse di hosting e ti offre un maggiore controllo sul modo in cui i giocatori vengono inseriti nelle sessioni di gioco.

- Sviluppa funzionalità di servizio di backend per l'hosting. Il servizio di backend comunica con il Amazon GameLift Servers servizio e fornisce informazioni di connessione a un client di gioco. Questa funzionalità include l'avvio di sessioni di gioco, l'inserimento di giocatori in partite e il recupero delle informazioni sulla sessione di gioco. Per le linee guida, consulta Integra giochi con server di gioco personalizzati. Effettua almeno quanto segue:
  - Scarica l' AWS SDK per Amazon GameLift Servers e aggiungilo al tuo progetto di servizio di backend. Consulta le risorse Amazon GameLift Servers SDK per i servizi client.
  - Aggiungi codice per inizializzare un Amazon GameLift Servers client e memorizzare le impostazioni chiave. Consultare Configura Amazon GameLift Servers su un servizio di backend.
  - Aggiungi funzionalità per richiamare l'azione AWS SDK `CreateGameSession()` e fornire informazioni sulla connessione della sessione di gioco a un client di gioco. Vedi Creare una sessione di gioco su una flotta specifica.

    `CreateGameSession()`La chiamata è un comodo punto di partenza per richiedere nuove sessioni di gioco. Dopo aver implementato un sistema di posizionamento delle sessioni di gioco (vedi Passaggio 3), sostituirai questo codice con una chiamata a `StartGameSessionPlacement()` (o `StartMatchmaking()` se stai utilizzandoFlexMatch).

    Per indicazioni sulla progettazione del servizio di backend, consulta. Progetta il tuo servizio client di gioco
- Aggiungi funzionalità al tuo client di gioco che consentano ai giocatori di partecipare a una sessione di gioco ospitata. Il client di gioco invia richieste al tuo servizio di backend, non direttamente aAmazon GameLift Servers. Dopo che il servizio di backend ha fornito le informazioni sulla connessione alla sessione di gioco, il client di gioco si connette direttamente alla sessione di gioco per giocare.
- Verifica l'integrazione del tuo client di gioco. Puoi utilizzare la stessa flotta Amazon GameLift Servers Anywhere con una workstation locale per i test.

Per uno sviluppo iterativo rapido o quando lavori con team composti da più persone, ti consigliamo di configurare un ambiente di test basato sul cloud. Questa soluzione Amazon GameLift Servers Toolkit imita il comportamento di una flotta Amazon GameLift Servers gestita ma consente di aggiornare le build dei server di gioco con tempi di risposta minimi.

Passaggio 3: imposta il posizionamento della sessione di gioco

Personalizza il modo in cui desideri Amazon GameLift Servers elaborare le richieste per una nuova sessione di gioco e individua i server di gioco disponibili per ospitarle. Amazon GameLift Serversmonitora automaticamente la disponibilità di tutti i server di gioco su tutte le flotte. Quando un client di gioco invia una richiesta di partecipazione a una sessione di gioco, Amazon GameLift Servers cerca il posizionamento «migliore possibile» in base a una serie di priorità definite come latenza minima, costo e disponibilità.

- Crea una coda di sessione di gioco per inserire una nuova sessione di gioco con i server di gioco disponibili. Le code sono il meccanismo principale per il posizionamento delle sessioni di gioco. Per le linee guida, consulta Crea una coda per le sessioni di gioco.
  - Come minimo, aggiungi le tue flotte Anywhere come destinazioni nella coda. Tutte le altre impostazioni sono personalizzazioni opzionali.
- Nel codice del servizio di backend, converti la **`CreateGameSession()`** chiamata in. **`StartGameSessionPlacement()`** Vedi Creare una sessione di gioco in una coda con più sedi.
- Crea un meccanismo per avvisare un client di gioco quando una sessione di gioco è pronta per partecipare. Durante lo sviluppo, puoi verificare lo stato della sessione di gioco utilizzando una chiamata a DescribeGameSessionPlacement. Prima di utilizzare una coda per elaborare volumi elevati, tuttavia, dovrai abilitare le notifiche degli eventi. Consultare Imposta la notifica degli eventi per il posizionamento della sessione di gioco.
- (Facoltativo) Aggiungi componenti di FlexMatch matchmaking. Per informazioni, consulta la guida per gli Amazon GameLift ServersFlexMatch sviluppatori.

Fase 4: Crea flotte gestite basate sul cloud

Fino a questo punto, hai lavorato con flotte Anywhere autogestite per testare e perfezionare i componenti del gioco e hai ottimizzato il posizionamento delle sessioni di gioco.L'ultima parte della soluzione consiste nel configurare il tipo di risorse di hosting necessarie per un sistema di

produzione. Per iniziare a pianificare e configurare la produzione, è necessario passare a lavorare con una flotta Amazon GameLift Servers gestita.

- Package della build del tuo server di gioco e caricala suAmazon GameLift Servers. Crea uno script di installazione con i tuoi file di build, le dipendenze e il software di supporto. Consultare Implementa una build di server personalizzata per Amazon GameLift Servers hosting. Puoi caricare la tua build Amazon GameLift Servers utilizzando la console o la AWS CLI.

  Prima di caricare la build, decidi in cosa Regione AWS vuoi creare una flotta. Devi caricare la build nella stessa regione. Per ulteriori informazioni sulla scelta dell'ubicazione della flotta, consultaUbicazione della flotta.
- Crea una EC2 flotta gestita. Quando crei una flotta, inizia Amazon GameLift Servers immediatamente a distribuire il server di gioco creato per l'hosting. Puoi configurare molti aspetti di una flotta gestita. Per le linee guida, consulta Crea una EC2 flotta Amazon GameLift Servers gestita. Come minimo, procedi come segue:
  - Assegna un nome alla flotta e specifica quale build di gioco caricata utilizzare.
  - Scegli le istanze On-Demand per il tuo parco istanze e seleziona un tipo di istanza disponibile in base all'ubicazione del tuo parco istanze. Le flotte Spot sono un'opzione preziosa, ma richiedono un design e una configurazione aggiuntivi.
  - Crea una configurazione di runtime per la flotta. Specificate almeno il percorso di avvio per il file eseguibile del server di gioco.
  - Specificate le impostazioni delle porte per consentire al traffico in entrata di accedere ai server di gioco.
- Aggiungi le flotte gestite alla tua coda. Nella coda delle sessioni di gioco, sostituisci le flotte Anywhere con le flotte gestite.
- Prova l'hosting di giochi con le tue flotte gestite. A questo punto dovresti essere in grado di testare l'intero ciclo di hosting, con un client di gioco che richiede una sessione di gioco, ottiene informazioni di connessione e si connette correttamente a una sessione di gioco.

Fase 5: Personalizza le flotte gestite

Mentre ti prepari per il lancio del gioco, dovrai ottimizzare le tue risorse di hosting gestito. Alcune delle decisioni da prendere in considerazione includono:

- Prendi in considerazione l'aggiunta di flotte Spot per risparmiare sui costi. Consultare Tutorial: creare una Amazon GameLift Servers coda con le istanze Spot.

- Se il tuo server di gioco deve comunicare altre AWS risorse, configura i ruoli IAM per gestire l'accesso. Consultare Comunica con altre AWS risorse delle tue flotte.
- Determina dove vuoi posizionare geograficamente i server di gioco. Aggiungi postazioni remote alle tue flotte gestite. Consultare Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite.
- Ottimizza le prestazioni del parco macchine selezionando il tipo e la dimensione dell'istanza e configurando il runtime per eseguire più processi server. Consultare Gestisci come Amazon GameLift Servers avvia i server di gioco.
- Sperimenta le opzioni di posizionamento delle sessioni di gioco per le flotte gestite, inclusa la personalizzazione delle impostazioni di prioritizzazione. Consultare Personalizza una coda di sessioni di gioco.
- Imposta il ridimensionamento automatico della capacità per soddisfare la domanda prevista dei giocatori. Consultare Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers.
- Configura flotte di standby in altri Regioni AWS e modifica le code e la scalabilità automatica per gestire i failover, se necessario.
- Configura gli strumenti di osservabilità dell'hosting, tra cui analisi e registrazione. Consultare Monitoraggio Amazon GameLift Servers.
- Automatizza la distribuzione utilizzando Infrastructure as Code (IaC). Consultare Gestione di Amazon GameLift Servers hosting di risorse utilizzando AWS CloudFormation.

  Amazon GameLift Serverssupporta l'uso di AWS CloudFormation modelli per qualsiasi configurazione specifica dell'implementazione. È inoltre possibile utilizzare il per definire le risorse. AWS Cloud Development Kit (AWS CDK) Amazon GameLift Servers Per ulteriori informazioni su AWS CDK, consulta la Guida per AWS Cloud Development Kit (AWS CDK) gli sviluppatori.

  Per gestire l'implementazione degli AWS CloudFormation stack, consigliamo di utilizzare strumenti e servizi di integrazione continua e distribuzione continua (CI/CD) come. AWS CodePipeline Questi strumenti ti aiutano a implementarlo automaticamente o con l'approvazione ogni volta che crei un file binario per server di gioco. Con CI/CD uno strumento o un servizio, la distribuzione delle risorse per una nuova versione del server di gioco può essere simile a questa:

  - Crea e testa il file binario del tuo server di gioco.
  - Carica il file binario suAmazon GameLift Servers.
  - Implementa nuove flotte con la nuova build.
  - Aggiungi le nuove flotte alla coda delle sessioni di gioco e rimuovi le flotte con la versione di build precedente.

- Quando le flotte della build precedente non ospitano più sessioni di gioco attive, elimina le AWS CloudFormation pile di quelle flotte.

Roadmap di sviluppo per l'hosting con contenitori Amazon GameLift Servers gestiti

Questa tabella di marcia ti guida nello sviluppo di una soluzione di hosting Amazon GameLift Servers gestito per i tuoi server di gioco containerizzati. I container gestiti sono solo una soluzione di hosting offerta da. Amazon GameLift Servers Per ulteriori informazioni sulle opzioni di hosting, consultaAmazon GameLift Serverssoluzioni.

Una soluzione container Amazon GameLift Servers gestita con i seguenti componenti:

- Una o più flotte di container, che utilizzano istanze Amazon Elastic Compute Cloud EC2 (Amazon) ottimizzate per l'hosting di giochi multiplayer.
- Un'immagine del contenitore con la build del server di gioco, caricata nell'archivio privato di Amazon Elastic Container Registry (Amazon ECR). La build del server di gioco è integrata con l'SDK del server per Linux Amazon GameLift Servers e progettata per funzionare su Linux.
- Un servizio di backend che interagisce con il Amazon GameLift Servers servizio per conto dei tuoi client di gioco. Il servizio di backend utilizza le funzionalità dell'API di servizio forAmazon GameLift Servers, che fa parte dell'SDK. AWS
- Una coda di sessioni di Amazon GameLift Servers gioco che elabora le richieste di nuove sessioni di gioco, cerca i server di gioco disponibili in tutte le flotte e richiede a un server di gioco di avviare una sessione di gioco.
- (Facoltativo) Un FlexMatch matchmaker per creare partite multigiocatore e impostare sessioni di gioco per esse.

Questa tabella di marcia presenta un percorso semplificato per rendere operativi con successo i server di gioco containerizzati con container gestiti. Amazon GameLift Servers Dopo aver installato i componenti necessari, puoi continuare a sviluppare giochi e personalizzare la tua soluzione di hosting. Man mano che ti avvicini al lancio, consulta questi articoli Preparazione del gioco per il lancio con Amazon GameLift Servers hosting per aiutarti a preparare la tua soluzione di hosting per l'utilizzo a livello di produzione.

Accelera l'onboarding con questi strumenti per container gestiti:

- Lo starter kit per container semplifica l'integrazione e la configurazione della flotta. Aggiunge funzionalità essenziali per la gestione delle sessioni di gioco al server di gioco e utilizza modelli preconfigurati per creare una flotta di container e una pipeline di distribuzione automatizzata per il server di gioco. Dopo l'implementazione, utilizza la Amazon GameLift Servers console e gli strumenti API per monitorare le prestazioni della flotta, gestire le sessioni di gioco e analizzare le metriche.
- Per gli sviluppatori di Unreal Engine e Unity, usa i Amazon GameLift Serversplugin per integrare il tuo server di gioco e creare una flotta di container dall'interno dell'ambiente di sviluppo del tuo motore di gioco. I flussi di lavoro guidati del plug-in ti aiutano a creare una soluzione semplice e veloce con l'hosting basato su cloud utilizzando contenitori gestiti. Puoi basarti su queste basi per creare una soluzione di hosting personalizzata per il tuo gioco.

Passaggio 1: prepara il server di gioco su cui lavorare Amazon GameLift Servers

Aggiungi funzionalità al tuo server di gioco in modo che possa comunicare con il Amazon GameLift Servers servizio quando viene distribuito per l'hosting.

- Scarica l'SDK del server Amazon GameLift Servers (versione 5.2 o successiva) per il tuo progetto di gioco. L'SDK del server è disponibile in C++, C# e Go. Scarica l'SDK del server per Amazon GameLift Servers. L'SDK del server è disponibile in C++, C# e Go.
- Modifica il codice del server di gioco per aggiungere la funzionalità SDK del server. Per le linee guida, consulta Integra giochi con server di gioco personalizzati. Come minimo, procedi come segue:
  - Aggiungi codice per inizializzare l'Amazon GameLift ServersSDK e stabilire una WebSocket connessione con il Amazon GameLift Servers servizio. Usa l'azione SDK del server. `InitSdk()`
  - Aggiungi codice da segnalare al Amazon GameLift Servers servizio quando il processo del server è pronto per ospitare sessioni di gioco. Usa l'azione `ProcessReady()` SDK del server.
  - Implementa le funzioni di callback richieste `OnStartGameSession()` e. `OnProcessTerminate()` Con queste funzioni, i processi del server di gioco possono mantenere una connessione con il Amazon GameLift Servers servizio, avviare una sessione di gioco quando richiesto e rispondere a una richiesta di Amazon GameLift Servers terminare il processo del server di gioco.

  - Aggiungi codice per segnalare al Amazon GameLift Servers servizio quando il processo del server sta terminando una sessione di gioco. Usa l'azione `ProcessEnding()` SDK del server.
- Package della build del tuo server di gioco. Crea il tuo server di gioco per funzionare su Linux. Prepara la build e gli altri file necessari per eseguire il server di gioco. Se stai sviluppando su Windows, questo passaggio potrebbe comportare la configurazione di un'area di lavoro Linux separata o l'utilizzo di uno strumento come Windows subsystem for Linux (WSL). Avrai bisogno di un ambiente Linux per testare la build del tuo server di gioco e anche per creare e testare le immagini dei tuoi container.
- Metti alla prova l'integrazione del tuo server di gioco. Verifica che il server di gioco integrato sia in grado di connettersi al Amazon GameLift Servers servizio e rispondere alle richieste. Ti consigliamo di configurare una semplice flotta Amazon GameLift Servers Anywhere con una workstation locale come host di test, come descritto in. Configura i test locali con Amazon GameLift Servers Anywhere Installa il tuo server di gioco, costruisci sull'host di test e avvia un processo server. Utilizza la AWS CLI per richiedere una nuova sessione di gioco e verifica che il Amazon GameLift Servers servizio richieda correttamente al processo del server di avviare una sessione di gioco.

Passaggio 2: prepara il client di gioco per partecipare alle sessioni di gioco ospitate

Crea un modo per consentire al client di gioco di richiedere di partecipare a una sessione di gioco, ottenere informazioni di connessione e quindi connettersi direttamente a una sessione di gioco ospitata. L'approccio più comune consiste nel configurare la funzionalità del servizio di backend che funga da intermediario tra il client di gioco e il servizio. Amazon GameLift Servers Questo approccio protegge le tue risorse di hosting e ti offre un maggiore controllo sul modo in cui i giocatori vengono inseriti nelle sessioni di gioco.

- Sviluppa funzionalità di servizio di backend per l'hosting. Il servizio di backend comunica con il Amazon GameLift Servers servizio e fornisce informazioni di connessione a un client di gioco. Questa funzionalità include l'avvio di sessioni di gioco, l'inserimento di giocatori in partite e il recupero delle informazioni sulla sessione di gioco. Per le linee guida, consulta Integra giochi con server di gioco personalizzati. Effettua almeno quanto segue:
  - Scarica l' AWS SDK per Amazon GameLift Servers e aggiungilo al tuo progetto di servizio di backend. Consulta le risorse Amazon GameLift Servers SDK per i servizi client.
  - Aggiungi codice per inizializzare un Amazon GameLift Servers client e memorizzare le impostazioni chiave. Consultare Configura Amazon GameLift Servers su un servizio di backend.

  - Aggiungi funzionalità per richiamare l'azione dell' AWS SDK `CreateGameSession()` e fornire informazioni sulla connessione della sessione di gioco a un client di gioco. Vedi Creare una sessione di gioco su una flotta specifica.

    `CreateGameSession()`La chiamata è un comodo punto di partenza per richiedere nuove sessioni di gioco. Dopo aver implementato un sistema di posizionamento delle sessioni di gioco (vedi Passaggio 3), sostituirai questo codice con una chiamata a `StartGameSessionPlacement()` (o `StartMatchmaking()` se stai utilizzandoFlexMatch).

    Per indicazioni sulla progettazione del servizio di backend, consulta. Progetta il tuo servizio client di gioco

- Aggiungi funzionalità al tuo client di gioco che consentano ai giocatori di partecipare a una sessione di gioco ospitata. Il client di gioco invia richieste al tuo servizio di backend, non direttamente aAmazon GameLift Servers. Dopo che il servizio di backend ha fornito le informazioni sulla connessione alla sessione di gioco, il client di gioco si connette direttamente alla sessione di gioco per giocare.
- Verifica l'integrazione del tuo client di gioco. Puoi utilizzare la tua flotta Amazon GameLift Servers Anywhere esistente con una workstation locale per i test. Utilizza il nuovo servizio di backend per richiedere una nuova sessione di gioco e verifica che: (1) il Amazon GameLift Servers servizio richieda correttamente al processo del server di avviare una sessione di gioco e (2) un client di gioco possa connettersi alla sessione di gioco.

Passaggio 3: imposta il posizionamento della sessione di gioco

Personalizza il modo in cui desideri Amazon GameLift Servers elaborare le richieste per una nuova sessione di gioco e individua i server di gioco disponibili per ospitarle. Amazon GameLift Serversmonitora automaticamente la disponibilità di tutti i server di gioco su tutte le flotte. Quando un client di gioco invia una richiesta di partecipazione a una sessione di gioco, Amazon GameLift Servers cerca il posizionamento «migliore possibile» in base a una serie di priorità definite come latenza minima, costo e disponibilità.

- Crea una coda di sessione di gioco per inserire una nuova sessione di gioco con i server di gioco disponibili. Le code sono il meccanismo principale per il posizionamento delle sessioni di gioco. Per le linee guida, consulta Crea una coda per le sessioni di gioco.
  - Come minimo, aggiungi le tue flotte Anywhere come destinazioni nella coda. Tutte le altre impostazioni sono personalizzazioni opzionali.

- Nel codice del servizio di backend, converti la **`CreateGameSession()`** chiamata in. **`StartGameSessionPlacement()`** Vedi Creare una sessione di gioco in una coda con più sedi.
- Crea un meccanismo per avvisare un client di gioco quando una sessione di gioco è pronta per partecipare. Durante lo sviluppo, puoi verificare lo stato della sessione di gioco utilizzando una chiamata a`DescribeGameSessionPlacement`. Prima di utilizzare una coda per elaborare volumi elevati, tuttavia, dovrai abilitare le notifiche degli eventi. Consultare Imposta la notifica degli eventi per il posizionamento della sessione di gioco.
- Aggiungi FlexMatch matchmaking (opzionale). Crea un set di regole di matchmaking e crea una configurazione di matchmaking per funzionare con la coda della tua sessione di gioco. Per indicazioni sulla configurazione di un sistema di matchmaking, consulta la guida per gli sviluppatori. Amazon GameLift ServersFlexMatch
- Prova il sistema di posizionamento. Puoi utilizzare la tua flotta Amazon GameLift Servers Anywhere esistente con una workstation locale per i test. Utilizza il servizio di backend per richiedere una nuova sessione di gioco e verifica che il Amazon GameLift Servers servizio richieda correttamente al processo del server di avviare una sessione di gioco.

Passaggio 4: Crea un'immagine del contenitore del server di gioco

Dopo aver integrato con successo il server di gioco, crea un'immagine contenitore con il file eseguibile del server di gioco. Archivialo in un repository Amazon Elastic Container Registry (Amazon ECR) per utilizzarlo con. Amazon GameLift Servers Per istruzioni dettagliate, vedi Crea un'immagine del contenitore per Amazon GameLift Servers.

- Scarica il modello Dockerfile per un contenitore di server di gioco (fornito da). Amazon GameLift Servers Modifica il file per i file di build del tuo server di gioco.
- Crea un'immagine del contenitore del server di gioco. Lavorando in un ambiente Linux, usa lo strumento Docker per creare la tua immagine.
- Invia l'immagine del contenitore ad Amazon ECR. Crea un repository pubblico o privato in Amazon ECR, utilizzandolo Account AWS e Regione AWS dove intendi distribuire la tua flotta di container. Inviaci l'immagine del contenitore.
- Testa le immagini dei container utilizzando la tua flotta Anywhere (opzionale). Potresti voler testare le immagini dei container localmente prima di distribuirle su una flotta di container ospitata nel cloud. Puoi utilizzare la tua flotta Amazon GameLift Servers Anywhere esistente con una workstation locale per i test. Installa ed esegui il contenitore del server di gioco e verifica che: (1) il Amazon GameLift Servers servizio richieda correttamente al processo del server di avviare una sessione di gioco e (2) un client di gioco possa connettersi alla sessione di gioco.

Fase 5: Crea una flotta di container basata sul cloud

Fino a questo punto hai lavorato con una flotta Anywhere autogestita per testare e iterare i componenti del tuo gioco. L'ultima parte della soluzione consiste nel configurare le risorse di hosting basate sul cloud necessarie per un sistema di produzione. Per iniziare a pianificare e configurare la produzione, è necessario configurare una flotta di container Amazon GameLift Servers gestita e personalizzarla per la produzione.

- Crea definizioni di gruppi di contenitori. Le definizioni dei gruppi di contenitori descrivono l'architettura dei container per una flotta e identificano quali immagini di container distribuire. Consultare Creare una definizione di gruppo di container per una flotta di Amazon GameLift Servers container. Crea la definizione del gruppo di contenitori nello stesso Regione AWS luogo in cui sono archiviate le immagini dei contenitori. Per ulteriori informazioni sulla scelta dell'ubicazione della flotta, consultaUbicazione della flotta. Come minimo, procedi come segue:
  - Crea una definizione del gruppo di contenitori del server di gioco.
  - Aggiungi una definizione di contenitore con un'immagine del contenitore con la build del tuo server di gioco.
  - Configura un intervallo di porte per i processi del server di gioco del contenitore.
- Crea una flotta di container gestita. Quando crei una flotta, inizia Amazon GameLift Servers immediatamente a distribuire il server di gioco creato per l'hosting. Puoi configurare molti aspetti di una flotta gestita. Per le linee guida, consulta Crea un Amazon GameLift Servers flotta di container gestita. Come minimo, procedi come segue:
  - Imposta un ruolo di servizio AWS Identity and Access Management (IAM) per la flotta di container. Consultare Configurare un ruolo di servizio IAM per Amazon GameLift Servers.
  - Specificate la definizione del gruppo di contenitori del server di gioco da distribuire sulle istanze della flotta.
  - Usa i valori predefiniti, se disponibili, per tutti gli altri parametri. Amazon GameLift Serverscalcola alcuni parametri per una configurazione ottimale.
- Aggiungi le flotte di container alla tua coda. Nella coda della sessione di gioco, sostituisci la flotta di test Anywhere con la tua flotta di container gestita.
- Metti alla prova l'hosting di giochi con le tue flotte di container. A questo punto dovresti essere in grado di testare l'intera soluzione. Avvia un client di gioco e richiedi una sessione di gioco tramite il servizio di backend. Ottieni informazioni sulla connessione e connettiti a una sessione di gioco sulla flotta di container.

- Ripeti le implementazioni della tua flotta. Puoi aggiornare le definizioni dei gruppi di container e le configurazioni della flotta e quindi distribuire aggiornamenti alle flotte esistenti.

Fase 6: Personalizza la tua soluzione container gestita

Mentre ti prepari per il lancio del gioco, dovrai ottimizzare le tue risorse di hosting gestito. Alcune delle decisioni da prendere in considerazione includono:

- Ottimizza la configurazione della tua flotta di container. Consultare Personalizza una flotta di Amazon GameLift Servers container.
- Prendi in considerazione l'aggiunta di flotte Spot per risparmiare sui costi. Consultare Tutorial: creare una Amazon GameLift Servers coda con le istanze Spot.
- Se il tuo server di gioco deve comunicare altre AWS risorse, configura i ruoli IAM per gestire l'accesso. Consultare Comunica con altre AWS risorse delle tue flotte.
- Determina dove vuoi posizionare geograficamente i server di gioco. Aggiungi postazioni remote alle tue flotte gestite. Consultare Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite.
- Sperimenta le opzioni di posizionamento delle sessioni di gioco per le flotte gestite, inclusa la personalizzazione delle impostazioni di prioritizzazione. Consultare Personalizza una coda di sessioni di gioco.
- Imposta il ridimensionamento automatico della capacità per soddisfare la domanda prevista dei giocatori. Consultare Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers.
- Crea flotte in altri Regioni AWS e modifica le code e la scalabilità automatica per gestire i failover secondo necessità.
- Configura gli strumenti di osservabilità dell'hosting, tra cui analisi e registrazione. Consultare Monitoraggio Amazon GameLift Servers.
- Automatizza le implementazioni della tua flotta utilizzando Infrastructure as Code (IaC). Consultare Gestione di Amazon GameLift Servers hosting di risorse utilizzando AWS CloudFormation.

  Amazon GameLift Serverssupporta l'uso di AWS CloudFormation modelli per qualsiasi configurazione specifica dell'implementazione. È inoltre possibile utilizzare il per definire le risorse. AWS Cloud Development Kit (AWS CDK) Amazon GameLift Servers Per ulteriori informazioni su AWS CDK, consulta la Guida per AWS Cloud Development Kit (AWS CDK) gli sviluppatori.

  Per gestire l'implementazione degli AWS CloudFormation stack, consigliamo di utilizzare strumenti e servizi di integrazione continua e distribuzione continua (CI/CD) come. AWS CodePipeline Questi strumenti ti aiutano a implementarlo automaticamente o con l'approvazione ogni volta che crei un

file binario per server di gioco. Con CI/CD uno strumento o un servizio, la distribuzione delle risorse per una nuova versione del server di gioco può essere simile a questa:

- Crea e testa il file binario del tuo server di gioco.
- Carica il file binario suAmazon GameLift Servers.
- Implementa nuove flotte con la nuova build.
- Aggiungi le nuove flotte alla coda delle sessioni di gioco e rimuovi le flotte con la versione di build precedente.
- Quando le flotte della build precedente non ospitano più sessioni di gioco attive, elimina le AWS CloudFormation pile di quelle flotte.

Roadmap di sviluppo per l'hosting con Anywhere Amazon GameLift Servers

Questa tabella di marcia illustra come sviluppare una soluzione di hosting per il gioco multiplayer da utilizzare con le tue risorse (hardware locale o macchine virtuali). Amazon GameLift Serversoffre diverse opzioni di hosting di giochi; per ulteriori informazioni su queste opzioni, consulta. Amazon GameLift Serverssoluzioni

Con l'hosting Amazon GameLift Servers Anywhere, il tuo server di gioco è ospitato su risorse informatiche fornite e gestite da te. Puoi creare una flotta Anywhere con le configurazioni di cui hai bisogno e dislocata geograficamente ovunque si trovino i tuoi giocatori. Amazon GameLift Serversoffre le seguenti funzionalità per una flotta Anywhere:

- Gestisce per te il processo di posizionamento delle sessioni di gioco in base alla tua configurazione, tra cui:
  - Monitoraggio della disponibilità dei server di gioco nelle tue flotte Anywhere.
  - Elaborazione delle richieste di gioco dal servizio client di gioco e abbinamento delle richieste di gioco ai server disponibili.
  - Richiedere ai server di gioco delle flotte Anywhere di avviare sessioni di gioco.
  - Comunicazione dei dettagli di connessione ai client di gioco.
- Raccoglie le metriche delle prestazioni per il processo di posizionamento delle sessioni e le metriche di utilizzo per le sessioni di gioco e i giocatori.
- Supporta il set completo di funzionalità di FlexMatch matchmaking, in modo da poter creare un matchmaker e integrarlo con il sistema di posizionamento delle sessioni di gioco.

- Offre all'Amazon GameLift Serversagente la gestione delle principali attività di gestione degli host su una flotta Anywhere.
- Supporta la combinazione con flotte Amazon GameLift Servers gestite per una soluzione ibrida flessibile.

Una soluzione Amazon GameLift Servers Anywhere è composta dai seguenti componenti:

- Una o più flotte Amazon GameLift Servers Anywhere con risorse di hosting locali o di altro tipo, gestite con gli strumenti di configurazione, gestione e distribuzione esistenti. (Facoltativamente, puoi usare il.) AWS Systems Manager
- Una versione di server di gioco, integrata con l'SDK del server perAmazon GameLift Servers, da distribuire su tutte le flotte.
- Un client di gioco e un servizio di backend, integrato con l' AWS SDK, per interagire con il Amazon GameLift Servers servizio e richiedere sessioni di gioco.
- Una Amazon GameLift Servers coda per effettuare nuove sessioni di gioco con i server di gioco disponibili su tutte le flotte.
- (Facoltativo) Un FlexMatch matchmaker per creare partite multigiocatore e impostare sessioni di gioco per esse.

Questa tabella di marcia presenta un percorso semplificato per avviare e far funzionare con successo il gioco multiplayer con l'hosting Anywhere. Amazon GameLift Servers Dopo aver installato i componenti necessari, puoi continuare a sviluppare giochi e personalizzare la tua soluzione di hosting. Man mano che ti avvicini al lancio, consulta questi articoli Preparazione del gioco per il lancio con Amazon GameLift Servers hosting per aiutarti a preparare la tua soluzione di hosting per l'utilizzo a livello di produzione.

> **Inizia subito con il Amazon GameLift Servers plugin per Unreal Engine e Unity**
>
> Per una distribuzione più rapida, prova il Amazon GameLift Serversplug-in per Unreal Engine e Unity. Fornisce flussi di lavoro guidati per l'interfaccia utente per implementare rapidamente il server di gioco con una configurazione minima, in modo da poter provare i componenti del gioco in azione. Quindi puoi basarti su queste basi per creare una soluzione di hosting personalizzata per il tuo gioco. Per ulteriori dettagli, consulta Esplora con Amazon GameLift Servers plugin.

Passaggio 1: prepara il server di gioco su cui lavorare Amazon GameLift Servers

Aggiungi funzionalità al tuo server di gioco in modo che possa comunicare con il Amazon GameLift Servers servizio quando viene distribuito per l'hosting.

- Scarica l'SDK del server Amazon GameLift Servers (versione 5.x) per il tuo progetto di gioco. L'SDK del server è disponibile in C++, C# e Go. Scarica un SDK per Amazon GameLift Servers server.
- Modifica il codice del server di gioco per aggiungere la funzionalità SDK del server. Per le linee guida, consulta Integra giochi con server di gioco personalizzati. Come minimo, procedi come segue:
  - Aggiungi codice per inizializzare l'Amazon GameLift ServersSDK e stabilire una WebSocket connessione con il Amazon GameLift Servers servizio. Utilizza l'azione SDK del server `InitSdk()` e includi i parametri del server, necessari per una flotta Anywhere.
  - Aggiungi codice per segnalare al Amazon GameLift Servers servizio quando il processo del server è pronto per ospitare sessioni di gioco. Usa l'azione `ProcessReady()` SDK del server.
  - Implementa le funzioni di callback richieste `OnProcessTerminate()` e. `OnStartGameSession()` Con queste funzioni, i processi del server di gioco possono mantenere una connessione con il Amazon GameLift Servers servizio, avviare una sessione di gioco quando richiesto e rispondere alla richiesta di Amazon GameLift Servers terminare il processo del server di gioco.
  - Aggiungi codice per segnalare al Amazon GameLift Servers servizio quando il processo del server sta terminando una sessione di gioco. Usa l'azione `ProcessEnding()` SDK del server.
- Package della build del tuo server di gioco. Crea uno script di installazione con i tuoi file di build, le dipendenze e il software di supporto. Consultare Creare un pacchetto dei file della build di gioco. Ti consigliamo di utilizzare un bucket Amazon Simple Storage Service (Amazon S3) per archiviare le versioni della build del gioco.
- Testa la tua integrazione con i server di gioco. Per questa operazione, consigliamo di configurare una flotta Amazon GameLift Servers Anywhere per una workstation locale, come descritto inConfigura i test locali con Amazon GameLift Servers Anywhere. Per questo passaggio, installa manualmente la build del server di gioco sul dispositivo di prova e avvia un processo server. Utilizza la AWS CLI per richiedere una nuova sessione di gioco e verifica che il Amazon GameLift Servers servizio richieda correttamente al processo del server di avviare una sessione di gioco.

Passaggio 2: prepara il client di gioco per partecipare alle sessioni di gioco ospitate

Crea un modo per consentire al client di gioco di richiedere di partecipare a una sessione di gioco, ottenere informazioni di connessione e quindi connettersi direttamente a una sessione di gioco ospitata. L'approccio più comune consiste nel configurare la funzionalità del servizio di backend che funga da intermediario tra il client di gioco e il servizio. Amazon GameLift Servers Questo approccio protegge le tue risorse di hosting e ti offre un maggiore controllo sul modo in cui i giocatori vengono inseriti nelle sessioni di gioco.

- Sviluppa funzionalità di servizio di backend per l'hosting. Il servizio di backend comunica con il Amazon GameLift Servers servizio e fornisce informazioni di connessione a un client di gioco. Questa funzionalità include l'avvio di sessioni di gioco, l'inserimento di giocatori in partite e il recupero delle informazioni sulla sessione di gioco. Per le linee guida, consulta Integra giochi con server di gioco personalizzati. Effettua almeno quanto segue:
  - Scarica l' AWS SDK per Amazon GameLift Servers e aggiungilo al tuo progetto di servizio di backend. Consulta le risorse Amazon GameLift Servers SDK per i servizi client.
  - Aggiungi codice per inizializzare un Amazon GameLift Servers client e memorizzare le impostazioni chiave. Consultare Configura Amazon GameLift Servers su un servizio di backend.
  - Aggiungi funzionalità per richiamare l'azione AWS SDK `CreateGameSession()` e fornire informazioni sulla connessione della sessione di gioco a un client di gioco. Vedi Creare una sessione di gioco su una flotta specifica.

    `CreateGameSession()`La chiamata è un comodo punto di partenza per richiedere nuove sessioni di gioco. Dopo aver implementato un sistema di posizionamento delle sessioni di gioco (vedi Passaggio 3), sostituirai questo codice con una chiamata a `StartGameSessionPlacement()` (o `StartMatchmaking()` se stai utilizzandoFlexMatch).

    Per indicazioni sulla progettazione del servizio di backend, consulta. Progetta il tuo servizio client di gioco
- Aggiungi funzionalità al tuo client di gioco che consentano ai giocatori di partecipare a una sessione di gioco ospitata. Il client di gioco invia richieste al tuo servizio di backend, non direttamente aAmazon GameLift Servers. Dopo che il servizio di backend ha fornito le informazioni sulla connessione alla sessione di gioco, il client di gioco si connette direttamente alla sessione di gioco per giocare.
- Verifica l'integrazione del tuo client di gioco. Puoi utilizzare la stessa flotta Amazon GameLift Servers Anywhere con workstation locali per i test.

Passaggio 3: imposta il posizionamento della sessione di gioco

Personalizza il modo in cui desideri Amazon GameLift Servers elaborare le richieste per una nuova sessione di gioco e individua i server di gioco disponibili per ospitarle. Amazon GameLift Serversmonitora automaticamente la disponibilità di tutti i server di gioco su tutte le flotte. Quando un client di gioco invia una richiesta di partecipazione a una sessione di gioco, Amazon GameLift Servers cerca il posizionamento «migliore possibile» in base a una serie di priorità definite come latenza minima, costo e disponibilità.

- Crea una coda di sessione di gioco per inserire una nuova sessione di gioco con i server di gioco disponibili. Le code sono il meccanismo principale per il posizionamento delle sessioni di gioco. Per le linee guida, consulta Crea una coda per le sessioni di gioco.
  - Come minimo, aggiungi le tue flotte Anywhere come destinazioni nella coda. Tutte le altre impostazioni sono facoltative.
- Nel codice del servizio di backend, converti la **`CreateGameSession()`** chiamata in. **`StartGameSessionPlacement()`** Vedi Creare una sessione di gioco in una coda con più sedi.
- Crea un meccanismo per avvisare un client di gioco quando una sessione di gioco è pronta per partecipare. Durante lo sviluppo, puoi verificare lo stato della sessione di gioco utilizzando una chiamata a DescribeGameSessionPlacement. Prima di utilizzare una coda per elaborare volumi elevati, tuttavia, dovrai abilitare le notifiche degli eventi. Consultare Imposta la notifica degli eventi per il posizionamento della sessione di gioco.
- (Facoltativo) Aggiungi componenti di FlexMatch matchmaking. Per informazioni, consulta la guida per gli Amazon GameLift ServersFlexMatch sviluppatori.

Fase 4: Configura una flotta Anywhere con l'Amazon GameLift Serversagente

Fino a questo punto hai utilizzato dispositivi locali (registrati come Anywhere fleet computes) per testare e iterare i componenti del gioco. Il passaggio successivo consiste nel configurare il tipo di flotta di cui avrai bisogno per un sistema di produzione. Per queste risorse, utilizza l'Amazon GameLift Serversagente per gestire alcune attività chiave di gestione degli host in ambito informatico. Per ulteriori dettagli, consulta Collabora con l'Amazon GameLift Serversagente.

- Scarica l'Amazon GameLift Serversagente e aggiungilo al pacchetto di installazione del server di gioco. Ottieni e crea il codice sorgente dell'agente, disponibile nel repository Amazon GameLift ServersAgent Github. Posiziona il file JAR risultante eseguibile nella stessa directory dell'eseguibile della build del gioco.

- Se necessario, modificate lo script di avvio per l'agente. Assicurati che l'eseguibile dell'agente venga avviato non appena un computer inizia a funzionare. Consulta il file readme nel repository Agent per informazioni sull'installazione e l'esecuzione dell'agente sui computer di hosting. Il comando di avvio dovrebbe includere opzioni per specificare, come minimo, l'Anywhere fleet ID e Regione AWS una posizione personalizzata e un nome di calcolo.

  L'agente gestisce automaticamente le seguenti attività per te, quindi se hai gestito queste attività con script, puoi rimuoverle:

  - Chiamate `RegisterCompute()` per aggiungere l'elaborazione a una flotta Anywhere.
  - Chiamate `GetComputeAuthToken()` per autenticare i server di gioco quando si connettono al Amazon GameLift Servers servizio. L'agente gestisce il recupero e l'aggiornamento del token di autenticazione, che può essere utilizzato da tutti i processi del server di gioco in esecuzione sul computer.
  - Avvia nuovi processi del server sul computer in base a una serie di istruzioni di runtime.
- Crea una configurazione di runtime per i computer della tua flotta Anywhere. È possibile utilizzare la Amazon GameLift Servers console o la AWS CLI per creare o modificare le istruzioni di runtime per la flotta. L'agente esegue queste istruzioni e richiede periodicamente aggiornamenti al Amazon GameLift Servers servizio.
- Configura o modifica la coda delle sessioni di gioco in base alle tue esigenze. Crea una nuova coda (o aggiornane una esistente) per utilizzare le flotte Anywhere distribuite con l'agente. Amazon GameLift Servers
- Testa l'integrazione di Agent con le tue flotte Anywhere. Verifica che l'agente stia avviando correttamente i processi del server in base alla configurazione di runtime.

Fase 5: Personalizza le tue flotte Anywhere

Mentre ti prepari per il lancio del gioco, dovrai ottimizzare le tue risorse di hosting gestito. Alcune delle decisioni da prendere in considerazione includono:

- Automatizza il processo di avvio e spegnimento dei computer in base alle esigenze, inclusa l'installazione e l'esecuzione del software del server di gioco. Il riciclaggio dei computer è utile per garantire che vengano aggiornati regolarmente e spegnerli può far risparmiare sui costi quando non sono necessari.
- Se il tuo server di gioco deve comunicare altre AWS risorse, configura i ruoli IAM per gestire l'accesso. Consultare Comunica con altre AWS risorse delle tue flotte.

- Determina dove vuoi posizionare geograficamente i server di gioco. Aggiungi postazioni remote alle tue flotte gestite. Consultare Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite.
- Ottimizza le prestazioni del parco macchine selezionando le configurazioni delle risorse di calcolo, quindi configura le istruzioni di runtime per eseguire un numero ottimale di processi server per elaborazione.
- Sperimenta le opzioni di posizionamento delle sessioni di gioco per flotte gestite, inclusa la personalizzazione delle impostazioni di prioritizzazione. Consultare Personalizza una coda di sessioni di gioco.
- Crea meccanismi per gestire la scalabilità manuale o automatica della capacità per soddisfare la domanda prevista dei giocatori. Considerate quali fattori dovrebbero indurre il sistema ad aumentare o diminuire il numero di computer disponibili per ospitare sessioni di gioco.
- Progetta e implementa il failover su altre risorse, se necessario.
- Configura strumenti di osservabilità dell'hosting, tra cui analisi e registrazione. Consultare Monitoraggio Amazon GameLift Servers.

Roadmap di sviluppo per l'hosting ibrido con Amazon GameLift Servers

Questa tabella di marcia ti guida nello sviluppo di una soluzione di hosting per il tuo gioco multiplayer. Amazon GameLift Serversoffre diverse opzioni di hosting di giochi; per ulteriori informazioni su queste opzioni, consultaAmazon GameLift Serverssoluzioni.

Una soluzione ibrida utilizza una combinazione di risorse di hosting, tra cui risorse basate sul cloud gestite da Amazon GameLift Servers e risorse di hosting gestite autonomamente dall'utente. Per una discussione più dettagliata sull'hosting ibrido, consulta questo articolo: Hosting di server di gioco ibridi con Anywhere. Amazon GameLift Servers ConAmazon GameLift Servers, puoi configurare una soluzione ibrida che utilizza componenti e processi comuni, in modo da gestire centralmente una flotta globale e spostare facilmente i carichi tra tutti i tipi di risorse.

Un'architettura ibrida è composta dai seguenti componenti:

- Una o più flotte Amazon GameLift Servers gestite, che utilizzano istanze Amazon Elastic Compute Cloud EC2 (Amazon) ottimizzate per l'hosting di giochi multiplayer.
- Una o più flotte Amazon GameLift Servers Anywhere, che utilizzano le tue risorse locali o di altro tipo di hosting esistenti, inclusi gli strumenti di configurazione, gestione e distribuzione. (Facoltativamente, puoi usare il.) AWS Systems Manager

- Una singola build di server di gioco, integrata con l'SDK del server perAmazon GameLift Servers, da distribuire su tutte le flotte.
- Un unico client di gioco e servizio di backend, integrato con l' AWS SDK, per interagire con il Amazon GameLift Servers servizio e richiedere sessioni di gioco.
- Una Amazon GameLift Servers coda condivisa per inserire nuove sessioni di gioco con i server di gioco disponibili e bilanciare il carico su tutte le flotte.
- The Amazon GameLift Servers Agent, che viene distribuito con una flotta Anywhere, per semplificare le attività di gestione dei processi dei server su tutti i computer di tutte le flotte.
- (Facoltativo) Un FlexMatch matchmaker per creare partite multigiocatore e impostare sessioni di gioco per esse.

Questa tabella di marcia presenta un percorso semplificato per avviare e far funzionare con successo il tuo gioco multiplayer in una soluzione di hosting ibrida con. Amazon GameLift Servers Dopo aver installato i componenti necessari, puoi continuare a sviluppare giochi e personalizzare la tua soluzione di hosting. Man mano che ti avvicini al lancio, consulta questi articoli Preparazione del gioco per il lancio con Amazon GameLift Servers hosting per aiutarti a preparare la tua soluzione di hosting per l'utilizzo a livello di produzione.

> **Inizia subito con il plugin Amazon GameLift Servers**
>
> Se stai sviluppando progetti con Unreal Engine o Unity, inizia a configurare il gioco per l'hosting con il Amazon GameLift Servers plug-in. Con il plug-in, puoi aggiungerlo Amazon GameLift Servers SDKs al tuo progetto di gioco e utilizzare i flussi di lavoro guidati per creare una versione semplice e funzionante di una soluzione di hosting ibrida con una flotta Anywhere e una flotta Amazon GameLift Servers gestita. Puoi quindi utilizzare questi principi fondamentali per svilupparli e personalizzarli secondo necessità.

Passaggio 1: prepara il server di gioco su cui lavorare Amazon GameLift Servers

Aggiungi funzionalità al tuo server di gioco in modo che possa comunicare con il Amazon GameLift Servers servizio quando viene distribuito per l'hosting. La stessa funzionalità è richiesta se il server di gioco è in esecuzione su una flotta Amazon GameLift Servers gestita o su una flotta Anywhere.

- Scarica l'SDK del server Amazon GameLift Servers (versione 5.x) per il tuo progetto di gioco. L'SDK del server è disponibile in C++, C# e Go. Scarica l'SDK del server per. Amazon GameLift Servers

- Modifica il codice del server di gioco per aggiungere la funzionalità SDK del server. Per le linee guida, consulta Integra giochi con server di gioco personalizzati. Come minimo, procedi come segue:
  - Aggiungi codice per inizializzare l'Amazon GameLift ServersSDK e stabilire una WebSocket connessione con il Amazon GameLift Servers servizio. Usa l'azione SDK del server. `InitSdk()` Includi il codice per specificare i parametri del server durante l'esecuzione su un computer Anywhere fleet.
  - Aggiungi codice per segnalare al Amazon GameLift Servers servizio quando il processo del server è pronto per ospitare sessioni di gioco. Usa l'azione `ProcessReady()` SDK del server.
  - Implementa le funzioni di callback richieste `OnProcessTerminate()` e. `OnStartGameSession()` Con queste funzioni, i processi del server di gioco possono mantenere una connessione con il Amazon GameLift Servers servizio, avviare una sessione di gioco quando richiesto e rispondere alla richiesta di Amazon GameLift Servers terminare il processo del server di gioco.
  - Aggiungi codice per segnalare al Amazon GameLift Servers servizio quando il processo del server sta terminando una sessione di gioco. Usa l'azione `ProcessEnding()` SDK del server.
- Package della build del tuo server di gioco. Crea uno script di installazione con i tuoi file di build, le dipendenze e il software di supporto. Consultare Creare un pacchetto dei file della build di gioco. Ti consigliamo di utilizzare un bucket Amazon Simple Storage Service (Amazon S3) per archiviare le versioni della build del gioco.
- Testa la tua integrazione con i server di gioco. Per questa operazione, consigliamo di configurare una flotta Amazon GameLift Servers Anywhere con una workstation locale, come descritto inConfigura i test locali con Amazon GameLift Servers Anywhere. Per questo passaggio, installa manualmente la build del server di gioco sul dispositivo di prova e avvia un processo server. Utilizza la AWS CLI per richiedere una nuova sessione di gioco e verifica che il Amazon GameLift Servers servizio richieda correttamente al processo del server di avviare una sessione di gioco.

Passaggio 2: prepara il client di gioco per partecipare alle sessioni di gioco ospitate

Crea un modo per consentire al client di gioco di richiedere di partecipare a una sessione di gioco, ottenere informazioni di connessione e quindi connettersi direttamente a una sessione di gioco ospitata. L'approccio più comune consiste nel configurare la funzionalità del servizio di backend che funga da intermediario tra il client di gioco e il servizio. Amazon GameLift Servers Questo approccio protegge le tue risorse di hosting e ti offre un maggiore controllo sul modo in cui i giocatori vengono inseriti nelle sessioni di gioco.

- Sviluppa funzionalità di servizio di backend per l'hosting. Il servizio di backend comunica con il Amazon GameLift Servers servizio e fornisce informazioni di connessione a un client di gioco. Questa funzionalità include l'avvio di sessioni di gioco, l'inserimento di giocatori in partite e il recupero delle informazioni sulla sessione di gioco. Per le linee guida, consulta Integra giochi con server di gioco personalizzati. Effettua almeno quanto segue:
  - Scarica l' AWS SDK per Amazon GameLift Servers e aggiungilo al tuo progetto di servizio di backend. Consulta le risorse Amazon GameLift Servers SDK per i servizi client.
  - Aggiungi codice per inizializzare un Amazon GameLift Servers client e memorizzare le impostazioni chiave. Consultare Configura Amazon GameLift Servers su un servizio di backend.
  - Aggiungi funzionalità per richiamare l'azione AWS SDK `CreateGameSession()` e fornire informazioni sulla connessione della sessione di gioco a un client di gioco. Vedi Creare una sessione di gioco su una flotta specifica.

    `CreateGameSession()`La chiamata è un comodo punto di partenza per richiedere nuove sessioni di gioco. Dopo aver implementato un sistema di posizionamento delle sessioni di gioco (vedi Passaggio 3), sostituirai questo codice con una chiamata a `StartGameSessionPlacement()` (o `StartMatchmaking()` se stai utilizzandoFlexMatch).

    Per indicazioni sulla progettazione del servizio di backend, consulta. Progetta il tuo servizio client di gioco
- Aggiungi funzionalità al tuo client di gioco che consentano ai giocatori di partecipare a una sessione di gioco ospitata. Il client di gioco invia richieste al tuo servizio di backend, non direttamente aAmazon GameLift Servers. Dopo che il servizio di backend ha fornito le informazioni sulla connessione alla sessione di gioco, il client di gioco si connette direttamente alla sessione di gioco per giocare.
- Verifica l'integrazione del tuo client di gioco. Puoi utilizzare la stessa flotta Amazon GameLift Servers Anywhere con una workstation locale per i test.

  Durante la fase di sviluppo, se desideri testare il comportamento della build del tuo gioco in una flotta Amazon GameLift Servers gestita, ti consigliamo di configurare anche un ambiente di test basato sul cloud. Questa soluzione Amazon GameLift Servers Toolkit imita il comportamento di una flotta gestita, ma consente di aggiornare le build dei server di gioco con tempi di risposta minimi.

Passaggio 3: imposta il posizionamento della sessione di gioco

Personalizza il modo in cui desideri Amazon GameLift Servers elaborare le richieste per una nuova sessione di gioco e individua i server di gioco disponibili per ospitarle. Amazon GameLift Serversmonitora automaticamente la disponibilità di tutti i server di gioco su tutte le flotte. Quando un client di gioco invia una richiesta di partecipazione a una sessione di gioco, Amazon GameLift Servers cerca il posizionamento «migliore possibile» in base a una serie di priorità definite come latenza minima, costo e disponibilità.

- Crea una coda di sessione di gioco per inserire una nuova sessione di gioco con i server di gioco disponibili. Le code sono il meccanismo principale per il posizionamento delle sessioni di gioco. Per le linee guida, consulta Crea una coda per le sessioni di gioco.
  - Come minimo, aggiungi le tue flotte Anywhere come destinazioni nella coda. Tutte le altre impostazioni sono personalizzazioni opzionali.
- Nel codice del servizio di backend, converti la **`CreateGameSession()`** chiamata in. **`StartGameSessionPlacement()`** Vedi Creare una sessione di gioco in una coda con più sedi.
- Crea un meccanismo per avvisare un client di gioco quando una sessione di gioco è pronta per partecipare. Durante lo sviluppo, puoi verificare lo stato della sessione di gioco utilizzando una chiamata a DescribeGameSessionPlacement. Prima di utilizzare una coda per elaborare volumi elevati, tuttavia, dovrai abilitare le notifiche degli eventi. Consultare Imposta la notifica degli eventi per il posizionamento della sessione di gioco.
- (Facoltativo) Aggiungi componenti di FlexMatch matchmaking. Per informazioni, consulta la guida per gli Amazon GameLift ServersFlexMatch sviluppatori.

Fase 4: Configura una flotta Anywhere con l'Amazon GameLift Serversagente

Fino a questo punto hai utilizzato dispositivi locali (registrati come Anywhere fleet computes) per testare e iterare i componenti del gioco. Il passaggio successivo consiste nel configurare il tipo di flotte necessarie per un sistema di produzione. Inizia con una flotta Anywhere e aggiungi l'Amazon GameLift ServersAgent per gestire alcune attività chiave di gestione degli host on-computing. Per ulteriori dettagli, consulta Collabora con l'Amazon GameLift Serversagente.

- Scarica l'Amazon GameLift Serversagente e aggiungilo al pacchetto di installazione del server di gioco. Ottieni e crea il codice sorgente dell'agente, disponibile nel repository Amazon GameLift ServersAgent Github. Posiziona il file JAR risultante eseguibile nella stessa directory dell'eseguibile della build del gioco.

- Se necessario, modificate lo script di avvio per l'agente. Assicurati che l'eseguibile dell'agente venga avviato non appena un computer inizia a funzionare. Consulta il file readme nel repository Agent per informazioni sull'installazione e l'esecuzione dell'agente sui computer di hosting. Il comando di avvio dovrebbe includere opzioni per specificare, come minimo, l'Anywhere fleet ID e Regione AWS una posizione personalizzata e un nome di calcolo.

  L'agente gestisce automaticamente le seguenti attività per te, quindi se hai gestito queste attività con script, puoi rimuoverle:

  - Chiamate `RegisterCompute()` per aggiungere l'elaborazione a una flotta Anywhere.
  - Chiamate `GetComputeAuthToken()` per autenticare i server di gioco quando si connettono al Amazon GameLift Servers servizio. L'agente gestisce il recupero e l'aggiornamento del token di autenticazione, che può essere utilizzato da tutti i processi del server di gioco in esecuzione sul computer.
  - Avvia nuovi processi del server sul computer in base a una serie di istruzioni di runtime.
- Crea una configurazione di runtime per i computer della tua flotta Anywhere. Specificate almeno il percorso di avvio per il file eseguibile del server di gioco. È possibile utilizzare la Amazon GameLift Servers console o la AWS CLI per creare o modificare le istruzioni di runtime per la flotta. L'agente esegue queste istruzioni e richiede periodicamente aggiornamenti al Amazon GameLift Servers servizio.
- Configura o modifica la coda delle sessioni di gioco in base alle tue esigenze. Crea una nuova coda (o aggiornane una esistente) e designa una destinazione per la flotta Anywhere schierata con l'agente. Amazon GameLift Servers
- Testa l'integrazione di Agent con le tue flotte Anywhere. Verifica che l'agente stia avviando correttamente i processi del server in base alla configurazione di runtime.

Fase 5: Creare una flotta gestita basata sul cloud

Crea una EC2 flotta Amazon GameLift Servers gestita per integrare la tua flotta Anywhere. Se configuri un ambiente di test basato sul cloud nella Fase 2 per accelerare lo sviluppo, pianifica di creare una flotta gestita dopo aver completato la maggior parte dello sviluppo e dei test del gioco. È necessaria una flotta completamente gestita per configurare e testare impostazioni aggiuntive come il ridimensionamento automatico della capacità.

- Package della build del tuo server di gioco e caricala suAmazon GameLift Servers. Crea uno script di installazione con i tuoi file di build, le dipendenze e il software di supporto. Puoi utilizzare lo stesso software di build sia con la tua flotta Anywhere che con quella gestita. Consultare

Implementa una build di server personalizzata per Amazon GameLift Servers hosting. Puoi caricare la tua build Amazon GameLift Servers utilizzando la console o la AWS CLI.

Prima di caricare la build, decidi in che Regione AWS modo vuoi creare la flotta gestita. Devi caricare la build nella stessa regione. Per ulteriori informazioni sulla scelta dell'ubicazione della flotta, consultaUbicazione della flotta.

- Crea una EC2 flotta gestita. Puoi utilizzare la Amazon GameLift Servers console o la AWS CLI per creare una flotta gestita. Quando crei una flotta, inizia Amazon GameLift Servers immediatamente a distribuire il server di gioco creato per l'hosting. Puoi configurare molti aspetti di una flotta gestita. Per le linee guida, consulta Crea una EC2 flotta Amazon GameLift Servers gestita. Come minimo, procedi come segue:
  - Assegna un nome alla flotta e specifica quale build di gioco caricata utilizzare.
  - Scegli le istanze On-Demand per il tuo parco istanze e seleziona un tipo di istanza disponibile in base all'ubicazione del tuo parco istanze. Le flotte Spot sono un'opzione preziosa, ma richiedono un design e una configurazione aggiuntivi.
  - Crea una configurazione di runtime con impostazioni simili a quelle utilizzate con la flotta Anywhere. Specificate almeno il percorso di avvio per il file eseguibile del server di gioco.
  - Specificate le impostazioni delle porte per consentire al traffico in entrata di accedere ai server di gioco.
- Aggiungi la flotta gestita alla coda delle sessioni di gioco condivise. Aggiorna la coda dalla Fase 4 in modo che includa le destinazioni sia per la flotta gestita che per la flotta Anywhere schierata con l'agente. Amazon GameLift Servers
- Prova l'hosting dei giochi con le tue flotte gestite. A questo punto dovresti essere in grado di testare l'intero ciclo di hosting, con un client di gioco che richiede una sessione di gioco, ottiene informazioni di connessione e si connette correttamente a una sessione di gioco.

Fase 6: Personalizza le tue flotte

Mentre ti prepari per il lancio del gioco, dovrai perfezionare le tue soluzioni di hosting. Alcune delle decisioni da prendere in considerazione includono:

- Per le flotte Anywhere, automatizza il processo di avvio e spegnimento dei computer in base alle esigenze, inclusa l'installazione e l'esecuzione del software del server di gioco. Il riciclaggio dei computer è utile per garantire che vengano aggiornati regolarmente e lo spegnimento dei computer può far risparmiare sui costi quando non sono necessari.

- Se il tuo server di gioco deve comunicare altre AWS risorse, configura i ruoli IAM per gestire l'accesso. Consultare Comunica con altre AWS risorse delle tue flotte.
- Determina dove vuoi posizionare geograficamente i server di gioco. Aggiungi postazioni remote alle tue flotte gestite. Consultare Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite.
- Per le flotte gestite, prendi in considerazione l'utilizzo delle flotte Spot per risparmiare sui costi. Consultare Tutorial: creare una Amazon GameLift Servers coda con le istanze Spot.
- Ottimizza le prestazioni della flotta selezionando le configurazioni delle risorse di calcolo, quindi configura le istruzioni di runtime per eseguire il numero ottimale di processi server per elaborazione. Esegui questa operazione sia per le flotte Anywhere che per le flotte gestite. Consultare Gestisci come Amazon GameLift Servers avvia i server di gioco.
- Sperimenta le opzioni di posizionamento delle sessioni di gioco per le flotte gestite, inclusa la personalizzazione delle impostazioni di prioritizzazione. Consultare Personalizza una coda di sessioni di gioco.
- Per le flotte gestite, imposta la scalabilità automatica della capacità per soddisfare la domanda prevista dei giocatori. Consultare Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers.
- Per le flotte Anywhere, crea meccanismi per gestire la scalabilità manuale o automatizzata della capacità per soddisfare la domanda prevista dei giocatori.
- Progetta e implementa il failover su altre risorse, se necessario. Configura flotte di standby in altri Regioni AWS e modifica le code e la scalabilità automatica per gestire i failover, se necessario.
- Configura gli strumenti di osservabilità dell'hosting, tra cui analisi e registrazione. Consultare Monitoraggio Amazon GameLift Servers. Crea gruppi di metrici per aggregare le analisi per tutte le tue risorse di hosting.
- Automatizza la distribuzione utilizzando Infrastructure as Code (IaC). Consultare Gestione di Amazon GameLift Servers hosting di risorse utilizzando AWS CloudFormation.

  Amazon GameLift Serverssupporta l'uso di AWS CloudFormation modelli per qualsiasi configurazione specifica dell'implementazione. È inoltre possibile utilizzare il per definire le risorse. AWS Cloud Development Kit (AWS CDK) Amazon GameLift Servers Per ulteriori informazioni su AWS CDK, consulta la Guida per AWS Cloud Development Kit (AWS CDK) gli sviluppatori.

  Per gestire l'implementazione degli AWS CloudFormation stack, consigliamo di utilizzare strumenti e servizi di integrazione continua e distribuzione continua (CI/CD) come. AWS CodePipeline Questi strumenti ti aiutano a implementarlo automaticamente o con l'approvazione ogni volta che crei un

file binario per server di gioco. Con CI/CD uno strumento o un servizio, la distribuzione delle risorse per una nuova versione del server di gioco può essere simile a questa:

- Crea e testa il file binario del tuo server di gioco.
- Carica il file binario suAmazon GameLift Servers.
- Implementa nuove flotte con la nuova build.
- Aggiungi le nuove flotte alla coda delle sessioni di gioco e rimuovi le flotte con la versione di build precedente.
- Quando le flotte della build precedente non ospitano più sessioni di gioco attive, elimina le AWS CloudFormation pile di quelle flotte.

Preparazione dei giochi per Amazon GameLift Servers

Prepara i tuoi giochi multiplayer per essere ospitati su Amazon GameLift Servers. Integrare Amazon GameLift Servers funzionalità di hosting nei tuoi progetti di gioco e crea server di gioco e server client. Configura un ambiente di test ospitato per supportare lo sviluppo e il test iterativi rapidi di giochi.

Argomenti

Integra giochi con server di gioco personalizzati

Amazon GameLift Servers fornisce un set completo di strumenti per preparare i giochi multiplayer e i server di gioco personalizzati su cui eseguirli Amazon GameLift Servers. La Amazon GameLift Servers SDKs contengono le librerie necessarie per comunicare con client e server di gioco Amazon GameLift Servers. Per ulteriori informazioni su SDKs e dove trovarli, vediOttieni strumenti Amazon GameLift Servers di sviluppo.

Gli argomenti di questa sezione contengono istruzioni dettagliate su come aggiungere Amazon GameLift Servers funzionalità sul client di gioco e sul server di gioco prima della distribuzione Amazon GameLift Servers.

Argomenti

Interazioni client/server di gioco con Amazon GameLift Servers

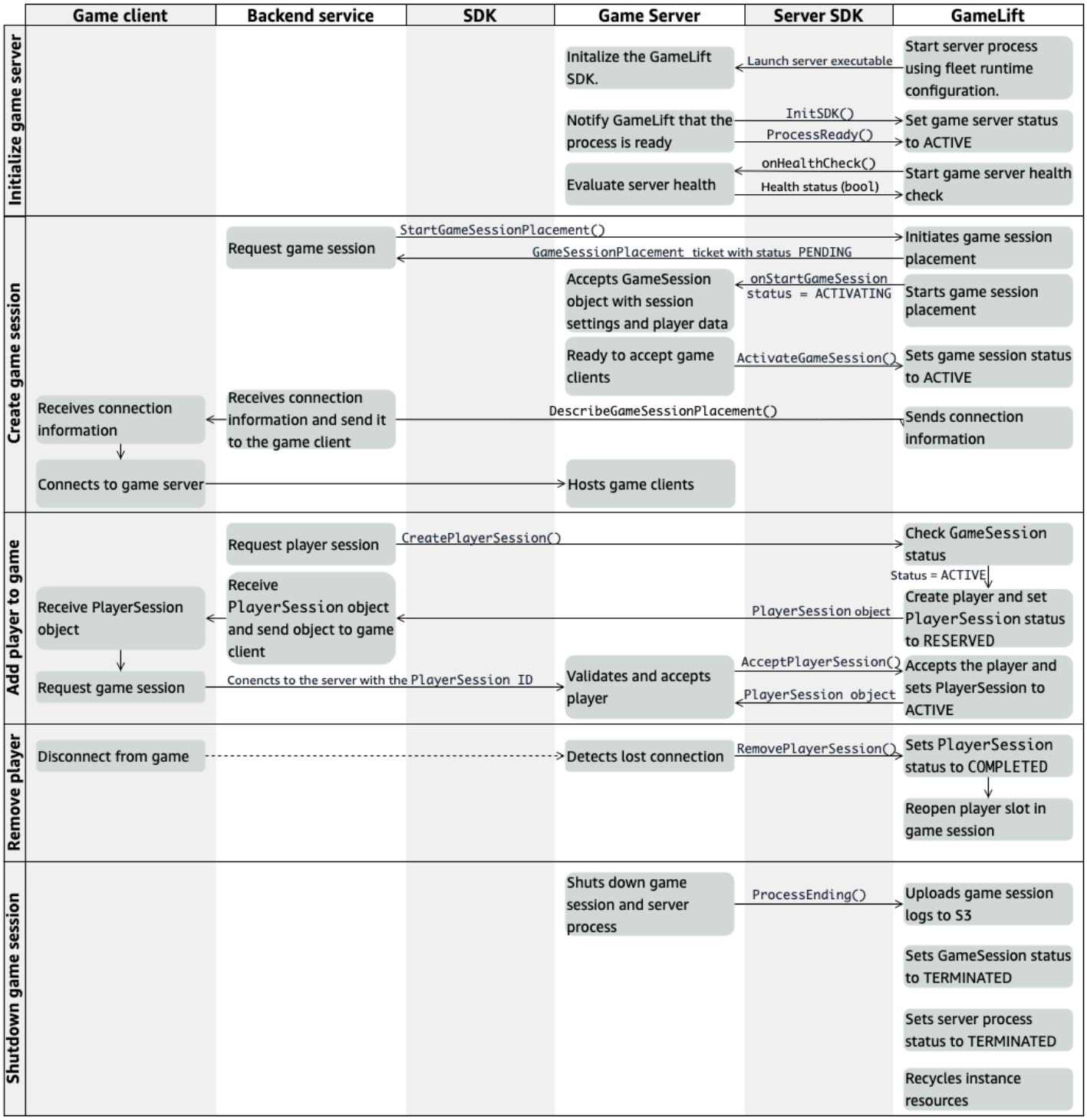
I componenti del tuo Amazon GameLift Servers le soluzioni di hosting interagiscono tra loro in modi specifici per eseguire sessioni di gioco in risposta alla domanda dei giocatori. Questo argomento descrive come i componenti comunicano tra loro quando il server di gioco è ospitato su Amazon GameLift Servers EC2 flotte gestite, autogestite Amazon GameLift Servers Flotte ovunque o soluzione ibrida.

I componenti della soluzione di hosting includono un server di gioco, il Amazon GameLift Servers servizio, un servizio di backend lato client e un client di gioco. Il server di gioco utilizza il Amazon GameLift Servers server SDK per interagire con Amazon GameLift Servers servizio. Il servizio di backend utilizza il Amazon GameLift Servers API di servizio (parte dell' AWS SDK) per interagire con il servizio per conto del client di gioco. Quando si accede a una sessione di gioco, il client di gioco si connette direttamente a una sessione di gioco utilizzando l'indirizzo IP e il numero di porta univoci della sessione di gioco.

Argomenti

Diagramma delle interazioni

Il diagramma seguente illustra come i componenti di hosting dei giochi interagiscono in modo che Amazon GameLift Servers il servizio può tenere traccia dello stato della disponibilità del server di gioco e avviare sessioni di gioco in risposta alle richieste dei giocatori.



Comportamenti di interazione

Le sezioni seguenti descrivono la sequenza di eventi in ciascuna delle interazioni chiave.

Inizializzazione di un processo del server di gioco

All'avvio, un processo del server di gioco stabilisce una comunicazione con Amazon GameLift Servers servizio e segnala il suo stato come pronto per ospitare una sessione di gioco.

1. Un nuovo processo dell'eseguibile del server di gioco inizia a funzionare su una risorsa di hosting.
2. Il processo del server di gioco richiama le seguenti operazioni SDK del server in sequenza:

   a. `InitSDK()`per inizializzare l'SDK del server, autenticare il processo del server e stabilire una comunicazione con Amazon GameLift Servers servizio.

   b. `ProcessReady()`per comunicare la disponibilità a ospitare una sessione di gioco. Questa chiamata riporta anche le informazioni di connessione del processo, quali client di gioco utilizzano per connettersi alla sessione di gioco e altre informazioni.

   Il processo del server attende quindi le richieste del Amazon GameLift Servers servizio.
3. Amazon GameLift Servers aggiorna lo stato del processo del server `ACTIVE` e lo rende disponibile per ospitare una nuova sessione di gioco.
4. Amazon GameLift Servers inizia a richiamare periodicamente il `onHealthCheck` callback per richiedere lo stato di integrità dei processi del server. Queste chiamate continuano mentre il processo del server rimane attivo. Il processo del server deve rispondere correttamente o meno entro un minuto. Se il processo del server non risponde correttamente o non risponde, a un certo punto il Amazon GameLift Servers il servizio modifica lo stato attivo del processo del server e interrompe l'invio di richieste per l'avvio della sessione di gioco.

Creare una sessione di gioco

Il Amazon GameLift Servers il servizio avvia una nuova sessione di gioco in risposta alla richiesta di un giocatore di giocare.

1. Un giocatore che utilizza il client di gioco chiede di partecipare a una sessione di gioco. A seconda di come il gioco gestisce la procedura di accesso dei giocatori, il client di gioco invia una richiesta al servizio di backend.
2. Se la procedura di iscrizione del giocatore richiede l'avvio di una nuova sessione di gioco, il servizio di backend invia una richiesta per una nuova sessione di gioco al Amazon GameLift Servers servizio. Questa richiesta richiama l'operazione dell'API del

servizio`StartGameSessionPlacement()`. (In alternativa, il servizio di backend potrebbe chiamare `StartMatchmaking()` o`CreateGameSession()`.)

3. Il Amazon GameLift Servers il servizio risponde creando un nuovo `GameSessionPlacement` ticket con stato. `PENDING` Restituisce le informazioni sul ticket al servizio di backend, in modo che possa tenere traccia dello stato del ticket di piazzamento e determinare quando la sessione di gioco è pronta per i giocatori. Per ulteriori informazioni, consulta Imposta la notifica degli eventi per il posizionamento della sessione di gioco.
4. Il Amazon GameLift Servers il servizio avvia il processo di posizionamento della sessione di gioco. Identifica le flotte da esaminare e cerca in esse un processo server attivo che non ospita una sessione di gioco. Individuando un processo server disponibile, il Amazon GameLift Servers il servizio esegue le seguenti operazioni:

   a. Crea un `GameSession` oggetto con le impostazioni della sessione di gioco e i dati del giocatore contenuti nella richiesta di posizionamento e imposta lo stato su`ACTIVATING`.

   b. Richiede al processo del server di avviare una sessione di gioco. Il servizio richiama il on`StartGameSession` callback del processo server e passa l'oggetto. `GameSession`

   c. Modifica il numero di sessioni di gioco del processo del server in. `1`
5. Il processo server esegue la sua funzione di on`StartGameSession` callback. Quando il processo server è pronto per accettare le connessioni dei giocatori, richiama l'operazione SDK del server `ActivateGameSession()` e attende le connessioni del giocatore.
6. Il Amazon GameLift Servers il servizio aggiorna l'`GameSession`oggetto con le informazioni di connessione per il processo del server (come riportato nella chiamata a`ProcessReady()`) e imposta lo stato della sessione di gioco su. `ACTIVE` Inoltre aggiorna lo stato `GameSessionPlacement` del ticket a`FULFILLED`.
7. Il servizio di backend chiama `DescribeGameSessionPlacement()` per controllare lo stato del ticket e ottenere informazioni sulla sessione di gioco. Quando la sessione di gioco è attiva, il servizio di backend notifica il client di gioco e trasmette le informazioni di connessione alla sessione di gioco.
8. Il client di gioco utilizza le informazioni di connessione per connettersi direttamente al processo del server di gioco e partecipare alla sessione di gioco.

Aggiungere un giocatore a una partita

Un gioco può opzionalmente utilizzare le sessioni dei giocatori per tenere traccia delle connessioni dei giocatori alle sessioni di gioco. Le sessioni dei giocatori possono essere create individualmente o come parte di una richiesta di posizionamento per le sessioni di gioco.

1. Il servizio di backend richiama l'operazione dell'API del servizio `CreatePlayerSession()` con un ID di sessione di gioco.
2. Il Amazon GameLift Servers il servizio controlla lo stato della sessione di gioco (deve esserlo`ACTIVE`) e cerca uno slot per giocatori libero nella sessione di gioco. Se è disponibile uno slot, il servizio esegue le seguenti operazioni:
    a. Crea un nuovo `PlayerSession` oggetto e imposta lo stato su`RESERVED`.
    b. Risponde alla richiesta del servizio di backend con informazioni sulla sessione del giocatore.
3. Il servizio di backend trasmette le informazioni sulla sessione del giocatore al client di gioco insieme alle informazioni sulla connessione della sessione di gioco.
4. Il client di gioco utilizza le informazioni di connessione e l'ID della sessione del giocatore per connettersi direttamente al processo del server di gioco e chiedere di partecipare alla sessione di gioco.
5. In risposta a un tentativo di accesso al client di gioco, il processo del server di gioco richiama l'API del servizio `AcceptPlayerSession()` per convalidare l'ID di sessione del giocatore. Il processo server accetta o rifiuta la connessione.
6. Il Amazon GameLift Servers il servizio esegue una delle seguenti operazioni:
    a. Se la connessione viene accettata, allora Amazon GameLift Servers imposta lo `PlayerSession` stato su `ACTIVE` e lo passa `PlayerSession` al processo del server di gioco.
    b. Se il processo del server di gioco non richiede `AcceptPlayerSession()` l'ID della sessione del giocatore entro un certo periodo di tempo dalla `CreatePlayerSession()` richiesta originale, Amazon GameLift Servers il servizio modifica lo `PlayerSession` stato `TIMEDOUT` e riapre lo slot del giocatore nella sessione di gioco.

Rimuovere un giocatore

Per i giochi che utilizzano sessioni di gioco, il processo del server di gioco notifica il Amazon GameLift Servers servizio quando un giocatore si disconnette. Il servizio utilizza queste informazioni

per tracciare lo stato degli slot dei giocatori in una sessione di gioco e può consentire ai nuovi giocatori di utilizzare gli slot aperti.

1. Un giocatore si disconnette dalla sessione di gioco.
2. Il processo del server di gioco rileva la connessione persa e richiama l'operazione SDK del server. `RemovePlayerSession()`
3. Il Amazon GameLift Servers il servizio modifica lo stato della sessione del giocatore `COMPLETED` e riapre lo slot del giocatore nella sessione di gioco.

Chiusura della sessione di gioco

Al termine di una sessione di gioco o quando si chiude la sessione di gioco, il processo del server notifica Amazon GameLift Servers servizio sullo stato della sessione di gioco.

1. Il processo del server di gioco termina la sessione di gioco e avvia l'arresto del processo richiamando l'operazione SDK del server. `ProcessEnding()`
2. Il Amazon GameLift Servers il servizio esegue le seguenti operazioni:
    a. Carica i log delle sessioni di gioco su Amazon Simple Storage Service (Amazon S3).
    b. Cambia lo stato della sessione di gioco in. `TERMINATED`
    c. Modifica lo stato del processo del server in`TERMINATED`.
    d. A seconda di come è progettata la soluzione di hosting, le nuove risorse di hosting disponibili vengono allocate per eseguire un nuovo processo del server di gioco.

Integra il tuo server di gioco con Amazon GameLift Servers

Dopo che il server di gioco personalizzato è stato distribuito e funzionato Amazon GameLift Servers in alcuni casi, deve essere in grado di interagire con Amazon GameLift Servers (e potenzialmente altre risorse). Questa sezione descrive come integrare il software del server di gioco con Amazon GameLift Servers.

> **Note**
>
> Queste istruzioni presuppongono che tu abbia creato un progetto di server di gioco Account AWS e che tu abbia già creato un progetto di server di gioco.

Gli argomenti di questa sezione descrivono come gestire le seguenti attività di integrazione:

- Stabilire una comunicazione tra Amazon GameLift Servers e i tuoi server di gioco.
- Genera e utilizza un certificato TLS per stabilire una connessione sicura tra il client di gioco e il server di gioco.
- Fornisci le autorizzazioni al software del tuo server di gioco per interagire con altre AWS risorse.
- Consenti ai processi dei server di gioco di ottenere informazioni sulla flotta su cui sono in esecuzione.

Argomenti

Aggiungi Amazon GameLift Servers al tuo server di gioco

Questo argomento descrive come modificare il codice del server di gioco in modo che i processi del server di gioco possano comunicare con il Amazon GameLift Servers servizio. Utilizza queste istruzioni per i server di gioco che intendi implementare su EC2 flotte Amazon GameLift Servers gestite, flotte di container gestite o flotte Anywhere.

I processi del server di gioco comunicano con il Amazon GameLift Servers servizio per ricevere istruzioni dal servizio e per segnalare lo stato dei processi del server e la sessione di gioco. Per informazioni dettagliate sulle interazioni tra i componenti della soluzione di hosting dei giochi (server di gioco, servizio di backend, client di gioco eAmazon GameLift Servers), consultaInterazioni client/server di gioco con Amazon GameLift Servers.

Per preparare il gioco per l'hosting, aggiungi l'SDK del server Amazon GameLift Servers al progetto del server di gioco. Se utilizzi il Amazon GameLift Servers plug-in per Unreal Engine o Unity, l'SDK del server è integrato e pronto per l'uso. Il Server SDK è disponibile in diverse lingue. Per ulteriori informazioni sul supporto degli strumenti per i server di gioco, incluso l'SDK del server, consulta. Ottieni strumenti Amazon GameLift Servers di sviluppo

Riferimenti alle API Server SDK:

- Server C++ SDK 5.x per -- Azioni Amazon GameLift Servers
- Server C# SDK 5.x per -- Azioni Amazon GameLift Servers
- Server C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers

- Go server SDK per Amazon GameLift Servers -- Azioni

Inizializza il processo del server

Aggiungi codice per stabilire una comunicazione con il Amazon GameLift Servers servizio e segnala quando il processo del server di gioco è pronto per ospitare una sessione di gioco. Questo codice deve essere eseguito prima di qualsiasi Amazon GameLift Servers codice.

1. Inizializza un client Amazon GameLift Servers API `InitSdk()` chiamando. Se stai preparando il server di gioco per l'esecuzione su EC2 flotte Amazon GameLift Servers gestite, usa l'impostazione predefinita `InitSDK()`(C++) (C#) (Unreal) (Go) (C++) senza parametri. Il client API gestisce la connessione al servizio per te. Amazon GameLift Servers

   > Se stai preparando il tuo server di gioco per l'uso su una flotta Amazon GameLift Servers Anywhere:
   >
   > Inizializza il client Amazon GameLift Servers API chiamando `InitSdk()` con quanto segue: `ServerParameters`
   >
   > - L'URL del websocket utilizzato per connettersi al server di gioco.
   > - L'ID del processo utilizzato per ospitare il server di gioco.
   > - L'ID del computer che ospita i processi del server di gioco.
   > - L'ID della flotta contenente il tuo computer Amazon GameLift Servers Anywhere.
   > - Il token di autorizzazione generato dall'Amazon GameLift Serversoperazione. `GetComputeAuthToken`

2. Notifica al servizio che il processo del server di gioco è pronto per ospitare una sessione di gioco. Chiama `ProcessReady()` (C++) (C#) (Unreal) (Go) (C++) con quanto segue. `ProcessParameters` Ogni processo del server di gioco deve chiamare solo una volta. `ProcessReady()`

   - Il numero di porta per il processo server. Quando il processo server avvia una sessione di gioco, fornisce la porta al Amazon GameLift Servers servizio, che aggiorna le informazioni sulla sessione di gioco. Il gioco può recuperare queste informazioni e fornirle ai client di gioco, che le utilizzano per connettersi al processo del server e partecipare alla sessione di gioco.
   - Le posizioni dei file che desideri Amazon GameLift Servers archiviare per te. Questi potrebbero includere i registri delle sessioni di gioco e altri file generati dal processo del server

durante una sessione di gioco. Sebbene Amazon GameLift Servers salvi temporaneamente questi file sul computer in cui è in esecuzione il processo del server, questi file sono disponibili solo fino alla chiusura dell'istanza. È possibile accedere ai file archiviati tramite la Amazon GameLift Serversconsole o chiamando l'operazione Amazon GameLift Servers API GetGameSessionLogUrl().

> Se stai preparando il tuo server di gioco per l'uso su una flotta di container Amazon GameLift Servers gestita:
>
> Non è necessario specificare i parametri di registro per una flotta di container. Invia invece la sessione di gioco e altri dati di registro allo standard output. Le flotte di container acquisiscono automaticamente tutti gli output standard dei container come flusso di log.

- Le seguenti funzioni di callback che consentono di Amazon GameLift Servers inviare messaggi o prompt a un processo del server di gioco. È necessario implementare ognuna di queste funzioni nel codice del server di gioco. Per ulteriori informazioni, consulta `ProcessParameters`(C++) (C#) (Unreal) (Go) (C++)Unreal).
  - (Facoltativo)`onHealthCheck`: Amazon GameLift Servers chiama questa funzione regolarmente per richiedere un rapporto sullo stato di salute del server.
  - `onStartGameSession`— Amazon GameLift Servers chiama questa funzione in risposta alla richiesta del client CreateGameSession().
  - `onProcessTerminate`— Amazon GameLift Servers forza l'arresto del processo del server, lasciandolo spegnersi correttamente.
  - (Facoltativo)`onUpdateGameSession`: Amazon GameLift Servers fornisce un oggetto della sessione di gioco aggiornato al server di gioco o fornisce un aggiornamento dello stato su una richiesta di ripristino della partita. La funzione di FlexMatchriempimento richiede questo callback.

Puoi anche configurare un server di gioco in modo che possa accedere in modo sicuro ad altre AWS risorse che possiedi o controlli. Per ulteriori informazioni, consulta Comunica con altre AWS risorse delle tue flotte.

(Facoltativo) Segnala lo stato dei processi del server

Aggiungi codice al tuo server di gioco per implementare la funzione `onHealthCheck()` di callback. Amazon GameLift Serversrichiama periodicamente questo metodo di callback per raccogliere metriche sullo stato di salute. Per implementare questa funzione di callback, procedi come segue:

- Valuta lo stato di salute del processo del server. Ad esempio, è possibile segnalare che il processo del server non è integro se alcune dipendenze esterne non sono riuscite.
- Completare la valutazione dello stato e rispondere alla richiamata entro 60 secondi. Se Amazon GameLift Servers non riceve una risposta entro tale lasso di tempo, considera automaticamente il processo del server non integro.
- Restituisce un valore booleano: true per salutare, false per non salutare.

Se non implementi un callback per il controllo dello stato di salute, Amazon GameLift Servers considera il processo del server integro a meno che il server non risponda.

Il Amazon GameLift Servers servizio utilizza l'integrità dei processi del server per porre fine a processi non integri e ripulire le risorse. Se un processo del server continua a essere segnalato come non integro o non risponde a tre controlli di integrità consecutivi, il servizio potrebbe arrestare il processo e avviarne uno nuovo. Il servizio raccoglie le metriche sullo stato dei processi del server di un parco macchine.

(Facoltativo) Ottieni un certificato TLS

Se il processo server è in esecuzione su una flotta in cui è attivata la generazione di certificati TLS, puoi recuperare il certificato TLS per stabilire una connessione sicura con un client di gioco e crittografare le comunicazioni client-server. Una copia del certificato viene archiviata nell'istanza. Per conoscere la posizione del file, chiamate `GetComputeCertificate()`(C++) (C#) (Unreal) (Go) (C++) (C#)Unreal).

Inizia una sessione di gioco

Aggiungi codice per implementare la funzione di callback. `onStartGameSession` Amazon GameLift Serversrichiama questo callback per avviare una sessione di gioco sul processo del server.

La `onStartGameSession` funzione accetta un GameSessionoggetto come parametro di input. Questo oggetto include informazioni chiave sulla sessione di gioco, come il numero massimo di giocatori. Può anche includere dati di gioco e dati dei giocatori. L'implementazione della funzione dovrebbe svolgere le seguenti attività:

- Avvia azioni per creare una nuova sessione di gioco in base alle `GameSession` proprietà. Come minimo, il server di gioco deve associare l'ID della sessione di gioco, a cui fanno riferimento i client di gioco quando si connettono al processo del server.
- Elabora i dati di gioco e i dati dei giocatori secondo necessità. Questi dati si trovano nell'`GameSession`oggetto.
- Avvisa il Amazon GameLift Servers servizio quando una nuova sessione di gioco è pronta per accettare giocatori. Chiama l'operazione dell'API del server `ActivateGameSession()`(C++) (C#) (Unreal) (Go) (C++)Unreal). In risposta a una chiamata riuscita, il servizio modifica lo stato della sessione di gioco in. `ACTIVE`

(Facoltativo) Convalida un nuovo giocatore

Se stai monitorando lo stato delle sessioni dei giocatori, aggiungi il codice per convalidare un nuovo giocatore quando si connette a un server di gioco. Amazon GameLift Serverstiene traccia dei giocatori attuali e degli slot disponibili per le sessioni di gioco.

Per la convalida, un client di gioco che tenta di partecipare a una sessione di gioco deve includere un ID di sessione del giocatore. Amazon GameLift Serversgenera questo ID quando il gioco inizia nuove sessioni di gioco chiamando StartGameSessionPlacement() o StartMatchmaking(). Su queste richieste, uno slot libero in una sessione di gioco è riservato alla sessione del giocatore.

Quando il processo del server di gioco riceve una richiesta di connessione al client di gioco, chiama `AcceptPlayerSession()` (C++) (C#) (Unreal) (Go) (C++) della sessione del giocatore. In risposta, Amazon GameLift Servers verifica che l'ID della sessione del giocatore corrisponda a uno slot aperto riservato nella sessione di gioco. Dopo aver Amazon GameLift Servers convalidato l'ID della sessione del giocatore, il processo del server accetta la connessione. Il giocatore può quindi partecipare alla sessione di gioco. Se Amazon GameLift Servers non convalida l'ID della sessione del giocatore, il processo del server nega la connessione.

(Facoltativo) Segnala la fine di una sessione di gioco

Se stai monitorando lo stato delle sessioni dei giocatori, aggiungi il codice per avvisare Amazon GameLift Servers quando un giocatore abbandona la sessione di gioco. Questo codice dovrebbe essere eseguito ogni volta che il processo del server rileva una connessione interrotta. Amazon GameLift Serversutilizza questa notifica per tenere traccia dei giocatori attuali e degli slot disponibili nella sessione di gioco.

Per gestire le connessioni interrotte nel codice, aggiungi una chiamata all'operazione dell'API del server `RemovePlayerSession()` (C++) (C#) (Unreal) (Go) (C++) di sessione del giocatore corrispondente.

Termina una sessione di gioco

Aggiungi codice alla sequenza di chiusura del processo del server per notificare Amazon GameLift Servers quando una sessione di gioco sta per terminare. Per riciclare e aggiornare le risorse di hosting, chiudi ogni processo del server al termine della sessione di gioco.

All'avvio del codice di spegnimento del processo server, richiama l'operazione dell'API del server `ProcessEnding()`(C++) (C#) (Unreal) (Go) g (C++) (C#)Unreal). Questa chiamata notifica Amazon GameLift Servers che il processo del server si sta chiudendo. Amazon GameLift Serversmodifica lo stato della sessione di gioco e lo stato del processo del server in. `TERMINATED` Dopo la chiamata`ProcessEnding()`, è sicuro che il processo si interrompa.

Rispondi a una notifica di chiusura di un processo del server

Aggiungi codice per arrestare il processo del server in risposta a una notifica del Amazon GameLift Servers servizio. Il servizio invia questa notifica quando il processo del server segnala costantemente che non è integro o se l'istanza in cui è in esecuzione il processo server viene interrotta. Amazon GameLift Serverspuò arrestare un'istanza come parte di un evento di riduzione della capacità o in risposta all'interruzione dell'istanza Spot. Un'interruzione di un'istanza Spot fornisce un preavviso di due minuti, che dà al server il tempo necessario per disconnettere correttamente i giocatori, conservare i dati sullo stato del gioco ed eseguire altre attività di pulizia.

Per gestire una notifica di spegnimento, apporta le seguenti modifiche al codice del server di gioco:

- Implementa la funzione di callback on`ProcessTerminate()`(C++) (C#) (Unreal) (Go) (C++) (C#). Questa funzione dovrebbe richiamare il codice che arresta il processo del server.
- Richiama l'operazione dell'API del server `GetTerminationTime()` (C++) (C#) (Unreal) (Go) (C++) dal codice di spegnimento del server di gioco. Se Amazon GameLift Servers ha emesso una chiamata per interrompere il processo del server, restituisce il tempo di terminazione stimato. `GetTerminationTime()`
- All'inizio del codice di spegnimento del server di gioco, richiama l'operazione API del server `ProcessEnding()`(C++) (C#) (Unreal) (Go) (C++) (C#). Questa chiamata avvisa il servizio Amazon GameLift Servers che il processo del server è in fase di arresto. Il servizio modifica quindi lo stato del processo del server in. `TERMINATED` Dopo la chiamata`ProcessEnding()`, è sicuro che il processo si interrompa.

Comunica con altre AWS risorse delle tue flotte

Quando crei una build di server di gioco da distribuire su Amazon GameLift Servers flotte, potresti volere che le applicazioni della build di gioco comunichino direttamente e in modo sicuro con altre AWS risorse di tua proprietà. Poiché Amazon GameLift Servers gestisce le tue flotte di giochi che ospitano giochi, devi concedere un accesso Amazon GameLift Servers limitato a queste risorse e servizi.

Alcuni scenari di esempio includono:

- Usa un CloudWatch agente Amazon per raccogliere metriche, log e tracce dalle EC2 flotte gestite e dalle flotte Anywhere.
- Invia i dati di log dell'istanza ad Amazon CloudWatch Logs.
- Ottieni file di gioco archiviati in un bucket Amazon Simple Storage Service (Amazon S3).
- Leggi e scrivi dati di gioco (come modalità di gioco o inventario) archiviati in un database Amazon DynamoDB o in un altro servizio di archiviazione dati.
- Invia segnali direttamente a un'istanza utilizzando Amazon Simple Queue Service (Amazon SQS).
- Accedi a risorse personalizzate distribuite e in esecuzione su Amazon Elastic Compute Cloud (Amazon EC2).

Amazon GameLift Serverssupporta questi metodi per stabilire l'accesso:

- Accedi alle AWS risorse con un ruolo IAM
- Accedi alle AWS risorse con il peering VPC

Accedi alle AWS risorse con un ruolo IAM

Utilizza un ruolo IAM per specificare chi può accedere alle tue risorse e impostare limiti a tale accesso. Le parti fidate possono «assumere» un ruolo e ottenere credenziali di sicurezza temporanee che le autorizzano a interagire con le risorse. Quando le parti effettuano richieste API relative alla risorsa, devono includere le credenziali.

Per configurare l'accesso controllato da un ruolo IAM, esegui le seguenti attività:

1. Crea il ruolo IAM
2. Modifica le applicazioni per acquisire credenziali

3. Associa una flotta al ruolo IAM

Crea il ruolo IAM

In questo passaggio, crei un ruolo IAM, con una serie di autorizzazioni per controllare l'accesso alle AWS risorse e una politica di fiducia che concede Amazon GameLift Servers i diritti di utilizzo delle autorizzazioni del ruolo.

Per istruzioni su come configurare il ruolo IAM, consulta. Configurare un ruolo di servizio IAM per Amazon GameLift Servers Quando crei la politica delle autorizzazioni, scegli servizi, risorse e azioni specifici con cui le tue applicazioni devono lavorare. Come procedura ottimale, limita il più possibile l'ambito delle autorizzazioni.

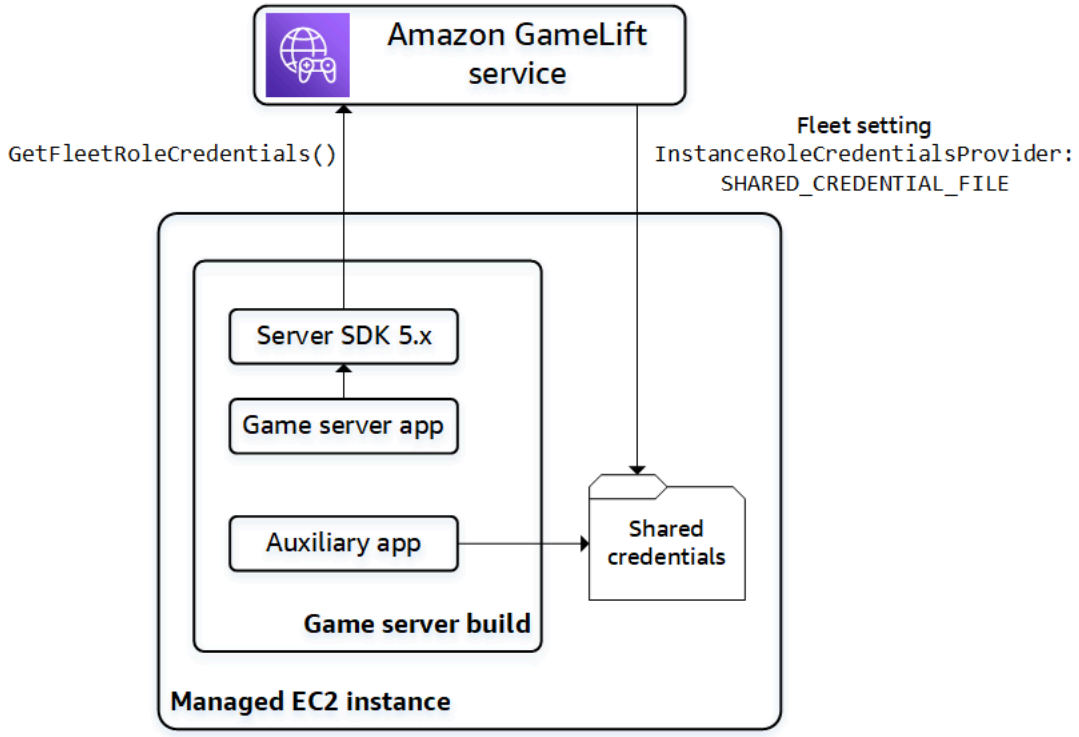
Dopo aver creato il ruolo, prendi nota dell'Amazon Resource Name (ARN) del ruolo. È necessario il ruolo ARN durante la creazione della flotta.

Modifica le applicazioni per acquisire credenziali

In questo passaggio, configuri le tue applicazioni per acquisire credenziali di sicurezza per il ruolo IAM e utilizzarle quando interagisci con le tue risorse. AWS Consulta la tabella seguente per determinare come modificare le applicazioni in base (1) al tipo di applicazione e (2) alla versione SDK del server con cui il gioco comunica. Amazon GameLift Servers

	Applicazioni per server di gioco	Altre applicazioni
Utilizzo della versione 5.x dell'SDK del server	Richiama il metodo SDK del server `GetFleetRoleCredentials()` dal codice del server di gioco.	Aggiungi codice all'applicazione per estrarre le credenziali da un file condiviso sull'istanza fleet.
Utilizzo della versione 4 o precedente dell'SDK del server	Chiama AWS Security Token Service (AWS STS) `AssumeRole` con il ruolo ARN.	Chiama AWS Security Token Service (AWS STS) `AssumeRole` con il ruolo ARN.

Per i giochi integrati con il server SDK 5.x, questo diagramma illustra come le applicazioni della build di gioco distribuita possono acquisire le credenziali per il ruolo IAM.



Chiamata **`GetFleetRoleCredentials()`** (server SDK 5.x)

Nel codice del server di gioco, che dovrebbe già essere integrato con il Amazon GameLift Servers server SDK 5.x, chiama `GetFleetRoleCredentials` (C++) (C#) (Unreal) (Go) per recuperare un set di credenziali temporanee. Quando le credenziali scadono, puoi aggiornarle con un'altra chiamata a. `GetFleetRoleCredentials`

Usa credenziali condivise (server SDK 5.x)

Per le applicazioni non server distribuite con build di server di gioco utilizzando server SDK 5.x, aggiungi codice per ottenere e utilizzare le credenziali archiviate in un file condiviso. Amazon GameLift Serversgenera un profilo di credenziali per ogni istanza del parco istanze. Le credenziali sono disponibili per l'uso da parte di tutte le applicazioni sull'istanza. Amazon GameLift Serversaggiorna continuamente le credenziali temporanee.

È necessario configurare una flotta per generare il file di credenziali condivise durante la creazione della flotta.

In ogni applicazione che deve utilizzare il file di credenziali condivise, specifica la posizione del file e il nome del profilo, come segue:

Windows:

```
[credentials]
shared_credential_profile= "FleetRoleCredentials"
shared_credential_file= "C:\\Credentials\\credentials"
```

Linux:

```
[credentials]
shared_credential_profile= "FleetRoleCredentials"
shared_credential_file= "/local/credentials/credentials"
```

Esempio: configura un CloudWatch agente per raccogliere le metriche per le istanze del parco istanze Amazon GameLift Servers

Se desideri utilizzare un CloudWatch agente Amazon per raccogliere metriche, log e tracce dalle tue Amazon GameLift Servers flotte, utilizza questo metodo per autorizzare l'agente a inviare i dati al tuo account. In questo scenario, procedi nel seguente modo:

1. Recuperate o scrivete il `config.json` file CloudWatch dell'agente.
2. Aggiornate il `common-config.toml` file per consentire all'agente di identificare il nome del file delle credenziali e il nome del profilo, come descritto sopra.
3. Configura lo script di installazione della build del server di gioco per installare e avviare l' CloudWatch agente.

Usa **`AssumeRole()`** (server SDK 4)

Aggiungi codice alle tue applicazioni per assumere il ruolo IAM e ottenere le credenziali per interagire con le tue AWS risorse. Qualsiasi applicazione eseguita su un'istanza del Amazon GameLift Servers parco istanze con server SDK 4 o versioni precedenti può assumere il ruolo IAM.

Nel codice dell'applicazione, prima di accedere a una AWS risorsa, l'applicazione deve chiamare l'operazione `AssumeRole` API AWS Security Token Service (AWS STS) e specificare il ruolo ARN. Questa operazione restituisce un set di credenziali temporanee che autorizzano l'applicazione ad accedere alla risorsa. AWS Per ulteriori informazioni, consulta Using temporary credenziali with AWS resources nella IAM User Guide.

Associa una flotta al ruolo IAM

Dopo aver creato il ruolo IAM e aggiornato le applicazioni nella build del server di gioco per ottenere e utilizzare le credenziali di accesso, puoi implementare una flotta. Quando configuri la nuova flotta, imposta i seguenti parametri:

- InstanceRoleArn— Imposta questo parametro sull'ARN del ruolo IAM.
- InstanceRoleCredentialsProvider— Amazon GameLift Servers Per richiedere la generazione di un file di credenziali condiviso per ogni istanza della flotta, imposta questo parametro su. `SHARED_CREDENTIAL_FILE`

È necessario impostare questi valori quando si crea la flotta. Non possono essere aggiornati in seguito.

Accedi alle AWS risorse con il peering VPC

Puoi utilizzare il peering di Amazon Virtual Private Cloud (Amazon VPC) per comunicare tra applicazioni in esecuzione su un'Amazon GameLift Serversistanza e un'altra risorsa. AWS Un VPC è una rete privata virtuale definita dall'utente che include un insieme di risorse gestite tramite. Account AWS Ogni parco istanze Amazon GameLift Servers ha il proprio VPC. Con il peering VPC, puoi stabilire una connessione di rete diretta tra il VPC per la tua flotta e per le altre risorse. AWS

Amazon GameLift Servers ottimizza il processo di configurazione delle connessioni VPC in peering per i server di gioco. Gestisce le richieste di peering, aggiorna le tabelle di routing e configura le connessioni in base alle esigenze. Per istruzioni su come configurare il peering VPC per i tuoi server di gioco, consulta. Peering VPC per Amazon GameLift Servers

Integra il tuo client di gioco conAmazon GameLift Servers

Gli argomenti di questa sezione descrivono il gestito Amazon GameLift Servers funzionalità che è possibile aggiungere a un servizio di backend. Un servizio di backend gestisce le seguenti attività:

- Richiede informazioni sulle sessioni di gioco attive a Amazon GameLift Servers.
- Inserisce un giocatore in una sessione di gioco esistente.
- Crea una nuova sessione di gioco e vi unisce i giocatori.
- Modifica i metadati di una sessione di gioco esistente.

Per ulteriori informazioni su come interagiscono i client di gioco con Amazon GameLift Servers e server di gioco in esecuzione su Amazon GameLift Servers, consulta Interazioni client/server di gioco con Amazon GameLift Servers.

Prerequisiti

- Un Account AWS.
- Una build del server di gioco caricata su Amazon GameLift Servers.
- Una flotta per ospitare i tuoi giochi.

Argomenti

Aggiungi Amazon GameLift Servers al tuo client di gioco

Amazon GameLift ServersIntegra componenti di gioco che richiedono informazioni sulla sessione di gioco, crea nuove sessioni di gioco e aggiungi giocatori ai giochi. A seconda dell'architettura di gioco, questa funzionalità è disponibile nei servizi di backend che gestiscono attività come l'autenticazione dei giocatori, il matchmaking o il posizionamento delle sessioni di gioco.

> **Note**
>
> Per informazioni dettagliate su come configurare il matchmaking per il tuo gioco, consulta la Guida per gli Amazon GameLift ServersFlexMatch sviluppatori.

Configura Amazon GameLift Servers su un servizio di backend

Aggiungi codice per inizializzare un Amazon GameLift Servers client e memorizzare le impostazioni chiave. Questo codice deve essere eseguito prima di qualsiasi codice dipendente Amazon GameLift Servers da.

1. Imposta una configurazione client. Utilizza la configurazione client predefinita o crea un oggetto di configurazione client personalizzato. Per ulteriori informazioni, vedere AWS::Client::ClientConfiguration(C++) o AmazonGameLiftConfig(C#).

Una configurazione client specifica una regione e un endpoint di destinazione da utilizzare per i contatti. Amazon GameLift Servers La regione identifica l'insieme di risorse distribuite (flotte, code e matchmaker) da utilizzare. La configurazione client predefinita imposta la posizione nella regione Stati Uniti orientali (Virginia settentrionale). Per utilizzare qualsiasi altra regione, crea una configurazione personalizzata.

2. Inizializzare un client Amazon GameLift Servers. Usa Aws:GameLift:: GameLiftClient () (C++) o AmazonGameLiftClient() (C#) con una configurazione client predefinita o una configurazione client personalizzata.
3. Aggiungi un meccanismo per generare un identificatore univoco per ogni giocatore. Per ulteriori informazioni, consulta Genera giocatore IDs.
4. Raccogli e archivia le seguenti informazioni:

   - Flotta target: molte richieste Amazon GameLift Servers API devono specificare un parco veicoli. A tale scopo, utilizza un ID flotta o un ID alias che punti alla flotta di destinazione. Come best practice, utilizza gli alias della flotta in modo da poter spostare i giocatori da una flotta all'altra senza dover aggiornare i servizi di backend.
   - Coda di destinazione: per i giochi che utilizzano code con più flotte per effettuare nuove sessioni di gioco, specifica il nome della coda da utilizzare.
   - AWS credenziali: tutte le chiamate a Amazon GameLift Servers devono fornire le credenziali dell'host del gioco. Account AWS L'utente acquisisce queste credenziali creando un utente giocatore, come descritto in. Configura l'accesso programmatico per il tuo gioco A seconda di come gestisci l'accesso per l'utente giocatore, procedi come segue:
     - Se utilizzi un ruolo per gestire le autorizzazioni utente del giocatore, aggiungi il codice per assumere il ruolo prima di chiamare un'Amazon GameLift ServersAPI. La richiesta di assunzione del ruolo restituisce una serie di credenziali di sicurezza temporanee. Per ulteriori informazioni, consulta Switching to an IAM role (AWS API) nella IAM User Guide.
     - Se disponi di credenziali di sicurezza a lungo termine, configura il codice per individuare e utilizzare le credenziali archiviate. Vedi Autenticazione tramite credenziali a lungo termine nella Guida di riferimento agli strumenti AWS SDKs e strumenti. Per informazioni sulla memorizzazione delle credenziali, consulta i riferimenti AWS API per (C++) e (.NET).
     - Se disponi di credenziali di sicurezza temporanee, aggiungi il codice per aggiornare regolarmente le credenziali utilizzando AWS Security Token Service (AWS STS), come descritto in Utilizzo delle credenziali di sicurezza temporanee con la Guida per l' AWS SDKsutente IAM. Il codice deve richiedere nuove credenziali prima della scadenza di quelle precedenti.

Ottieni sessioni di gioco

Aggiungi codice per scoprire le sessioni di gioco disponibili e gestire le impostazioni e i metadati delle sessioni di gioco.

Cerca sessioni di gioco attive

Consente SearchGameSessionsdi ottenere informazioni su una sessione di gioco specifica, su tutte le sessioni attive o sulle sessioni che soddisfano una serie di criteri di ricerca. Questa chiamata restituisce un GameSessionoggetto per ogni sessione di gioco attiva che corrisponde alla richiesta di ricerca.

Utilizza i criteri di ricerca per ottenere un elenco filtrato di sessioni di gioco attive a cui possono unirsi i giocatori. Ad esempio, è possibile filtrare le sessioni come segue:

- Escludi le sessioni di gioco complete:`CurrentPlayerSessionCount = MaximumPlayerSessionCount`.
- Scegli le sessioni di gioco in base alla durata della sessione: Valuta`CreationTime`.
- Trova sessioni di gioco in base a una proprietà di gioco personalizzata:`gameSessionProperties.gameMode = "brawl"`.

Gestisci le sessioni di gioco

Utilizzare una delle operazioni elencate di seguito per recuperare o aggiornare le informazioni sulla sessione di gioco.

- DescribeGameSessionDetails() — Visualizza lo stato di protezione di una sessione di gioco oltre alle informazioni sulla sessione di gioco.
- UpdateGameSession() — Modifica i metadati e le impostazioni di una sessione di gioco in base alle esigenze.
- GetGameSessionLogUrl— Accedi ai registri delle sessioni di gioco memorizzati.

Crea sessioni di gioco

Aggiungere codice per avviare nuove sessioni di gioco nei parchi di istanze distribuite e renderle disponibili ai giocatori. Esistono due opzioni per creare sessioni di gioco, a seconda che tu stia distribuendo il gioco in più Regioni AWS o in una singola regione.

Crea una sessione di gioco in una coda con più sedi

Si usa StartGameSessionPlacementper inserire una richiesta per una nuova sessione di gioco in una coda. Per utilizzare questa operazione, crea una coda. Questo determina dove Amazon GameLift Servers colloca la nuova sessione di gioco. Per ulteriori informazioni sulle code e su come utilizzarle, consultaGestione del posizionamento delle sessioni di gioco con Amazon GameLift Servers code.

Quando crei un posizionamento per una sessione di gioco, specifica il nome della coda da utilizzare, un nome per la sessione di gioco, un numero massimo di giocatori simultanei e un set opzionale di proprietà del gioco. Facoltativamente, puoi anche fornire un elenco di giocatori a cui partecipare automaticamente alla sessione di gioco. Se includi i dati sulla latenza dei giocatori per le regioni pertinenti, Amazon GameLift Servers utilizza queste informazioni per collocare la nuova sessione di gioco su una flotta che offra l'esperienza di gioco ideale per i giocatori.

Per ottenere misurazioni accurate della latenza, utilizza i beacon Amazon GameLift Servers di ping UDP. Questi endpoint consentono di misurare l'effettiva latenza della rete UDP tra i dispositivi dei giocatori e le potenziali sedi di hosting, con il risultato di decisioni di posizionamento più accurate rispetto all'utilizzo dei ping ICMP. Per ulteriori informazioni sull'utilizzo dei beacon ping UDP per misurare la latenza, fare riferimento a. Fari di ping UDP

Il posizionamento delle sessioni di gioco è un processo asincrono. Dopo aver effettuato una richiesta, puoi lasciarla andare a buon fine o scadere. Puoi anche annullare la richiesta in qualsiasi momento utilizzando StopGameSessionPlacement. Per verificare lo stato della tua richiesta di collocamento, chiama DescribeGameSessionPlacement.

Crea una sessione di gioco su una flotta specifica

Usa CreateGameSessionper creare una nuova sessione su una flotta specificata. Questa operazione sincrona riesce o non riesce a seconda che il parco disponga di risorse disponibili per ospitare una nuova sessione di gioco. Dopo aver Amazon GameLift Servers creato la nuova sessione di gioco e aver GameSessionrestituito un oggetto, puoi unirvi altri giocatori.

Quando usi questa operazione, fornisci un ID flotta o un ID alias, un nome di sessione e il numero massimo di giocatori simultanei per il gioco. In alternativa, è possibile includere un set di proprietà di gioco. Le proprietà del gioco sono definite in una serie di coppie chiave-valore.

Se utilizzi la funzionalità di protezione Amazon GameLift Servers delle risorse per limitare il numero di sessioni di gioco che un giocatore può creare, fornisci l'ID giocatore del creatore della sessione di gioco.

Unisciti a un giocatore a una sessione di gioco

Aggiungi codice per riservare uno slot per giocatore in una sessione di gioco attiva e connettere i client di gioco alle sessioni di gioco.

1. Riserva uno slot per giocatori in una sessione di gioco

   Per prenotare uno slot del giocatore, creare una nuova sessione del giocatore per la sessione di gioco. Per ulteriori informazioni sulle sessioni con i giocatori, consultaCome i giocatori si connettono ai giochi.

   Esistono due modi per creare nuove sessioni con i giocatori:

   - StartGameSessionPlacementUtilizzalo per riservare slot per uno o più giocatori nella nuova sessione di gioco.
   - Riserva gli slot per uno o più giocatori utilizzando CreatePlayerSessiono CreatePlayerSessionscon un ID di sessione di gioco.

   Amazon GameLift Serversverifica innanzitutto che la sessione di gioco accetti nuovi giocatori e che siano disponibili degli slot per giocatori. In caso di successo, Amazon GameLift Servers riserva uno slot per il giocatore, crea la nuova sessione di gioco e restituisce un PlayerSessionoggetto. Questo oggetto contiene il nome DNS, l'indirizzo IP e la porta di cui un client di gioco ha bisogno per connettersi alla sessione di gioco.

   La richiesta della sessione di un giocatore deve includere un ID univoco per ogni giocatore. Per ulteriori informazioni, consulta Genera giocatore IDs.

   Una sessione di gioco può includere una serie di dati personalizzati del giocatore. Questi dati vengono memorizzati nell'oggetto della sessione del giocatore appena creato, che puoi recuperare chiamando DescribePlayerSessions(). Amazon GameLift Serverspassa questo oggetto anche al server di gioco quando il giocatore si connette direttamente alla sessione di gioco. Quando richiedi sessioni con più giocatori, fornisci una stringa di dati per ogni giocatore mappata all'ID giocatore indicato nella richiesta.

2. Connect a una sessione di gioco

   Aggiungere codice al client di gioco per recuperare l'oggetto `PlayerSession` che contiene le informazioni di connessione della sessione di gioco. Usa queste informazioni per stabilire una connessione diretta al server.

- È possibile connettersi utilizzando la porta specificata e il nome DNS o l'indirizzo IP assegnato al processo del server.
- Se le tue flotte hanno abilitato la generazione di certificati TLS, connettiti utilizzando il nome e la porta DNS.
- Se il server di gioco convalida le connessioni dei giocatori in entrata, fai riferimento all'ID della sessione del giocatore.

Dopo aver effettuato la connessione, il client di gioco e il processo server comunicano direttamente senza coinvolgere. Amazon GameLift Servers Il server mantiene la comunicazione con Amazon GameLift Servers cui segnalare lo stato della connessione del giocatore, lo stato di salute e altro ancora. Se il server di gioco convalida i giocatori in arrivo, verifica che l'ID della sessione del giocatore corrisponda a uno slot riservato nella sessione di gioco e accetta o nega la connessione del giocatore. Quando il giocatore si disconnette, il processo del server segnala l'interruzione della connessione.

Usa le proprietà della sessione di gioco

Il client di gioco può trasferire dati a una sessione di gioco utilizzando una proprietà di gioco. Le proprietà del gioco sono coppie chiave-valore che il server di gioco può aggiungere, leggere, elencare e modificare. Puoi trasmettere una proprietà di gioco quando crei una nuova sessione di gioco o successivamente quando la sessione di gioco è attiva. Una sessione di gioco può contenere fino a 16 proprietà del gioco. Non puoi eliminare le proprietà del gioco.

Ad esempio, il gioco offre i seguenti livelli di difficoltà: `NoviceEasy`,`Intermediate`, e`Expert`. Un giocatore sceglie e `Easy` poi inizia la partita. Il client di gioco richiede una nuova sessione di gioco Amazon GameLift Servers utilizzando una delle due `StartGameSessionPlacement` opzioni o `CreateGameSession` come spiegato nelle sezioni precedenti. Nella richiesta, il client passa questo:`{"Key": "Difficulty", "Value":"Easy"}`.

In risposta alla richiesta, Amazon GameLift Servers crea un `GameSession` oggetto che contiene la proprietà di gioco specificata. Amazon GameLift Serversquindi ordina a un server di gioco disponibile di avviare la nuova sessione di gioco e passa l'`GameSession`oggetto. Il server di gioco avvia una sessione di gioco con un `Difficulty` di`Easy`.

Ulteriori informazioni

- GameProperty tipo di dati

- SearchGameSessions() esempi
- UpdateGameSession GameProperties parametro ()

Genera giocatore IDs

Amazon GameLift Servers utilizza una sessione giocatore per rappresentare un giocatore connesso a una sessione di gioco. Amazon GameLift Servers crea una sessione giocatore ogni volta che un giocatore si connette a una sessione di gioco utilizzando un client di gioco integrato con Amazon GameLift Servers. Quando un giocatore abbandona una partita, la sessione termina. Amazon GameLift Servers non riutilizza le sessioni dei giocatori.

Il seguente esempio di codice genera casualmente un giocatore unico: IDs

```
bool includeBrackets = false;
bool includeDashes = true;
string playerId = AZ::Uuid::CreateRandom().ToString<string>(includeBrackets,
 includeDashes);
```

Per ulteriori informazioni sulle sessioni con i giocatori, consulta. Sessioni di gioco e di gioco nella Amazon GameLift Servers console

Motori di gioco e Amazon GameLift Servers

Puoi usare il gestore Amazon GameLift Servers servizio con la maggior parte dei principali motori di gioco che supportano le librerie C++ o C#, tra cui O3DE, Unreal Engine e Unity. Crea la versione di cui hai bisogno per il gioco; consulta i file README di ciascuna versione per le istruzioni per la compilazione e i requisiti minimi. Per ulteriori informazioni, disponibile Amazon GameLift Servers SDKs, piattaforme di sviluppo e sistemi operativi supportati, vedi Ottieni strumenti Amazon GameLift Servers di sviluppo per i server di gioco.

Oltre alle informazioni specifiche sul motore fornite in questo argomento, trova ulteriore assistenza sull'integrazione Amazon GameLift Servers nei tuoi server di gioco, client e servizi nei seguenti argomenti:

- Aggiungi Amazon GameLift Servers al tuo server di gioco— Istruzioni dettagliate sull'integrazione Amazon GameLift Servers in un server di gioco.
- Aggiungi Amazon GameLift Servers al tuo client di gioco: istruzioni dettagliate sull'integrazione in un servizio o client di gioco, tra cui la creazione di sessioni di gioco e l'aggiunta di utenti ai giochi.

O3DE

Server di gioco

Prepara i tuoi server di gioco per l'hosting su Amazon GameLift Servers utilizzando l'SDK del server per Amazon GameLift Servers per C++. Per ottenere assistenza con l'integrazione della funzionalità desiderata nel server di gioco, consulta Aggiungi Amazon GameLift Servers al tuo server di gioco.

Client e servizi di gioco

Consenti ai tuoi client di gioco e/o servizi di gioco di interagire con Amazon GameLift Servers servizio, ad esempio per trovare sessioni di gioco disponibili o crearne di nuove e aggiungere giocatori ai giochi. Le funzionalità client di base sono fornite nell'AWS SDK for C++. Per integrare Amazon GameLift Servers nel tuo progetto di gioco O3DE, vedi Add (Aggiungi) Amazon GameLift Servers su un client e server di gioco O3DE e. Aggiungi Amazon GameLift Servers al tuo client di gioco

Unreal Engine

Server di gioco

Prepara i tuoi server di gioco per l'hosting su Amazon GameLift Servers aggiungendo l'SDK del server per Amazon GameLift Servers per Unreal Engine al tuo progetto e implementando le funzionalità del server richieste. Per assistenza nella configurazione del plug-in Unreal Engine e nell'aggiunta Amazon GameLift Servers codice, vediIntegrazione Amazon GameLift Servers in un progetto Unreal Engine.

Client e servizi di gioco

Consenti ai tuoi client di gioco e/o servizi di gioco di interagire con Amazon GameLift Servers servizio, ad esempio per trovare sessioni di gioco disponibili o crearne di nuove e aggiungere giocatori ai giochi. Le funzionalità client di base sono fornite nell'AWS SDK for C++. Per integrare Amazon GameLift Servers nel tuo progetto di gioco Unreal Engine, vediAggiungi Amazon GameLift Servers al tuo client di gioco.

Unità

Server di gioco

Prepara i server di gioco per l'hosting su Amazon GameLift Servers aggiungendo l'SDK del server per Amazon GameLift Servers per C# al progetto e implementando le funzionalità del server

richieste. Per assistenza nella configurazione di Unity e nell'aggiunta Amazon GameLift Servers codice, vediIntegrazione Amazon GameLift Servers in un progetto Unity.

Client e servizi di gioco

Consenti ai tuoi client di gioco e/o servizi di gioco di interagire con Amazon GameLift Servers servizio, ad esempio per trovare sessioni di gioco disponibili o crearne di nuove e aggiungere giocatori ai giochi. La funzionalità client principale è fornita da AWS SDK per .NET. Per integrare Amazon GameLift Servers nel tuo progetto di gioco Unity, vediAggiungi Amazon GameLift Servers al tuo client di gioco.

Altri motori

Per un elenco completo dei Amazon GameLift Servers SDKs disponibile per server e client di gioco, vedithe section called “Ottieni strumenti di sviluppo”.

Add (Aggiungi) Amazon GameLift Servers su un client e server di gioco O3DE

Puoi utilizzare O3DE, un motore 3D open source, multipiattaforma e in tempo reale per creare esperienze interattive ad alte prestazioni, inclusi giochi e simulazioni. Il renderer e gli strumenti O3DE sono racchiusi in un framework modulare che puoi modificare ed estendere con i tuoi strumenti di sviluppo preferiti.

Il framework modulare utilizza Gems che contengono librerie con interfacce e risorse standard. Seleziona le tue gemme per scegliere quali funzionalità aggiungere in base alle tue esigenze.

Il Amazon GameLift Servers Gem offre le seguenti funzionalità:

Amazon GameLift Servers integrazione

> Un framework per estendere il livello di rete O3DE e consentire a Multiplayer Gem di funzionare con Amazon GameLift Servers soluzione server dedicata. The Gem fornisce integrazioni sia con l'SDK del server che Amazon GameLift Serverse il client AWS SDK (per chiamare il Amazon GameLift Servers servizio stesso).

Gestione delle build e dei pacchetti

> Istruzioni per impacchettare e, facoltativamente, caricare la build del server dedicato e un'applicazione AWS Cloud Development Kit (AWS CDK) (AWS CDK) per configurare e aggiornare le risorse.

Amazon GameLift Servers Configurazione Gem

Segui le procedure in questa sezione per configurare Amazon GameLift Servers Gem in O3DE.

Prerequisiti

- Configura il tuo account per AWS Amazon GameLift Servers. Per ulteriori informazioni, vedereConfigura un Account AWS.
- Imposta AWS le credenziali per O3DE. Per ulteriori informazioni, consulta Configurazione delle credenziali. AWS
- Configura e. AWS CLI AWS CDK Per ulteriori informazioni, AWS Command Line Interfacee AWS Cloud Development Kit (AWS CDK).

Attiva il Amazon GameLift Servers Gem e le sue dipendenze

1. Apri il Project Manager.
2. Apri il menu sotto il tuo progetto e scegli Modifica impostazioni progetto... .
3. Scegli Configure Gems.
4. Attiva il Amazon GameLift Servers Gem e le seguenti gemme dipendenti:
   - AWS Core Gem: fornisce il framework da utilizzare Servizi AWS in O3DE.
   - Multiplayer Gem: fornisce funzionalità multiplayer estendendo il framework di rete.

Includi il Amazon GameLift Servers Libreria statica Gem

1. Includi l'`Gem::AWSGameLift.Server.Static`as come `BUILD_DEPENDENCIES` destinazione del tuo Project Server.

```
ly_add_target(
    NAME YourProject.Server.Static STATIC
    ...
    BUILD DEPENDCIES
        PUBLIC
            ...
        PRIVATE
            ...
            Gem::AWSGameLift.Server.Static
)
```

2. Impostato `AWSGameLiftService` come richiesto per il componente di sistema del Project Server.

```
void
 YourProjectServerSystemComponent::GetRequiredServices(AZ::ComponentDescriptor::DependencyA
 required)
{
    ...
    required.push_back(AZ_CRC_CE("AWSGameLiftServerService"));
    ...
}
```

3. (Facoltativo) Da creare Amazon GameLift Servers richieste di servizio in C++, incluse `Gem::AWSGameLift.Client.Static` nel target `BUILD_DEPENDENCIES` for your client.

```
ly_add_target(
    NAME YourProject.Client.Static STATIC
    ...
    BUILD_DEPENDENCIES
    PUBLIC
        ...
    PRIVATE
        ...
        Gem::AWSCore.Static
        Gem::AWSGameLift.Client.Static
}
```

Integra il gioco e il server dedicato

Gestisci le sessioni di gioco all'interno del tuo gioco e del server di gioco dedicato con l'integrazione della gestione delle sessioni. Per supportare FlexMatch, vedi FlexMatch Integrazione.

Integrazione Amazon GameLift Servers in un progetto Unreal Engine

Scopri come integrare l'Amazon GameLift ServersSDK per Unreal Engine nei tuoi progetti di gioco per accedere al set completo di funzionalità SDK del server.

> **Suggerimento**
>
> Per una distribuzione più rapida, prova il plug-in Amazon GameLift Servers standalone per Unreal Engine. Fornisce flussi di lavoro guidati per l'interfaccia utente per implementare

rapidamente il server di gioco con una configurazione minima, in modo da poter provare i componenti del gioco in azione. Consultare Amazon GameLift Serversplugin per Unreal Engine.

Risorse aggiuntive:

- Server C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers
- Ottieni strumenti Amazon GameLift Servers di sviluppo

Installa l'SDK del server per Unreal

Scarica l'Amazon GameLift ServersSDK open source per Unreal Engine da. GitHub I file readme del repository contengono i prerequisiti e le istruzioni di installazione.

Imposta gli obiettivi di compilazione e le regole del modulo

Modifica i file di progetto del gioco per generare correttamente i componenti di compilazione da utilizzare conAmazon GameLift Servers.

Per aggiungere obiettivi di build per client e server:

1. Apri i file di codice del progetto di gioco e individua il `.../Games/[your application name]Source/[your application name]Target.cs` file. Esempio: `.../Source/GameLiftUnrealAppTarget.cs`. (Se usi Visual Studio, apri il `.sln` file del progetto).
2. Copia questo file per creare due nuovi file di destinazione nella `Source/` directory.
   - Target client: rinomina il nuovo file in`[your application name]Client.Target.cs`. Modifica il contenuto per aggiornare i valori del nome della classe e del tipo di destinazione, come illustrato nel seguente codice di esempio:

```
using UnrealBuildTool;
  using System.Collections.Generic;

  public class GameLiftUnrealAppClientTarget :  TargetRules
 {
     public GameLiftUnrealAppClientTarget ( TargetInfo Target ) :  base
 ( Target )
     {
         Type = TargetType.Client;
```

```
        DefaultBuildSettings = BuildSettingsVersion.V2;
        IncludeOrderVersion = EngineIncludeOrderVersion.Unreal5_1;
        ExtraModuleNames.Add( "GameLiftUnrealApp");
    }
}
```

- Destinazione del server: rinomina il nuovo file in. *[your application name]*`Server.Target.cs` Modifica il contenuto per aggiornare i valori del nome della classe e del tipo di destinazione, come illustrato nel seguente codice di esempio:

```
using UnrealBuildTool;
 using System.Collections.Generic;

 public class GameLiftUnrealAppServerTarget :  TargetRules
{
    public GameLiftUnrealAppServerTarget ( TargetInfo Target ) :  base
( Target )
    {
        Type = TargetType.Server;
        DefaultBuildSettings = BuildSettingsVersion.V2;
        IncludeOrderVersion = EngineIncludeOrderVersion.Unreal5_1;
        ExtraModuleNames.Add( "GameLiftUnrealApp");
    }
}
```

3. Rigenera i file di progetto. Se usi Visual Studio, puoi fare clic con il pulsante destro del mouse sul `.uproject` file del progetto di gioco e selezionare Genera file di progetto di Visual Studio.

Per aggiornare le regole del modulo del progetto di gioco:

Aggiorna le regole del modulo del progetto di gioco per fare in modo che dipenda dal plugin.

1. Apri i file di codice del progetto di gioco e individua il `.../Games/`*[your application name]*`Source/`*[your application name]*`.Build.cs` file. Esempio: `.../Source/GameLiftUnrealApp.Build.cs`. (Se usi Visual Studio, apri il `.sln` file del progetto).
2. Individua la `ModuleRules` classe e aggiorna come illustrato nel seguente codice di esempio:

```
using UnrealBuildTool;

 public class GameLiftUnrealApp :  ModuleRules
{
```

```
    public GameLiftUnrealApp ( ReadOnlyTargetRules Target ) :  base ( Target )
    {
        PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;

        PublicDependencyModuleNames.AddRange( new string[] {  "Core",
"CoreUObject",  "Engine",  "InputCore",  "HeadMountedDisplay",  "EnhancedInput",
"GameLiftServerSDK" });

        bEnableExceptions =  true;
    }
}
```

3. Dopo aver creato i nuovi file di destinazione e modificato le regole del modulo, ricostruisci il progetto del gioco.

Aggiungi funzionalità di hosting di giochi al codice del tuo server

Dopo l'installazione e la configurazione dell'SDK del server, il passaggio successivo consiste nell'integrare la funzionalità di hosting dei giochi nel codice del server. L'SDK del server consente al server di gioco di comunicare con il Amazon GameLift Servers servizio, ricevere istruzioni per le sessioni di gioco, segnalare lo stato e lo stato di salute ed eseguire altre azioni.

Questo argomento fornisce un esempio di codice che aggiunge le funzionalità minime necessarie per ospitare il gioco. Amazon GameLift Servers

Passaggio 1: Aggiornare il file di **`GameMode`** intestazione

1. Apri i file di codice del progetto di gioco e individua il *`Your-application-name`*`GameMode.h` file. Esempio: `GameLiftUnrealAppGameMode.h`. Se usi Visual Studio, apri il `.sln` file relativo al progetto del gioco.
2. Modifica il file di intestazione per includere il seguente codice di esempio. Assicurati di sostituire "GameLiftUnrealApp" con il nome della tua applicazione.

Codice di esempio per GameMode .h

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

#pragma once
```

```
#include "CoreMinimal.h"
#include "GameFramework/GameModeBase.h"
#include "GameLiftUnrealAppGameMode.generated.h"

struct FProcessParameters;

DECLARE_LOG_CATEGORY_EXTERN(GameServerLog, Log, All);

UCLASS(minimalapi)
class AGameLiftUnrealAppGameMode : public AGameModeBase
{
    GENERATED_BODY()

public:
    AGameLiftUnrealAppGameMode();

protected:
    virtual void BeginPlay() override;

private:
    void InitGameLift();

private:
    TSharedPtr<FProcessParameters> ProcessParameters;
};
```

Passaggio 2: aggiungi le chiamate SDK del server richieste al codice del server di gioco

Usa il codice di esempio in questa sezione per integrare il codice del server di gioco da utilizzare conAmazon GameLift Servers. Per informazioni dettagliate sulle funzioni del codice, consulta Inizializza il processo del server eServer C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers.

> **Note**
>
> Il flag `WITH_GAMELIFT` del preprocessore ha due scopi:
>
> - Limita le chiamate API di Amazon GameLift Servers backend solo alle build del server Unreal
> - Garantisce la compatibilità tra i diversi obiettivi di build di Unreal

1. Apri il *Your-application-name*GameMode.cpp file sorgente correlato. Nel nostro esempio:GameLiftUnrealAppGameMode.cpp.
2. Modifica il codice per allinearlo al seguente codice di esempio. Assicurati di sostituire qualsiasi istanza di "GameLiftUnrealApp" con il nome della tua applicazione.

   L'esempio di codice fornito mostra come aggiungere gli elementi richiesti per l'integrazione conAmazon GameLift Servers. Ciò include:

   - Inizializza un client Amazon GameLift Servers API.
   - Implementa funzioni di callback per rispondere alle richieste del Amazon GameLift Servers servizio, tra cui OnStartGameSessionOnProcessTerminate, e. onHealthCheck
   - Chiama ProcessReady() per avvisare il Amazon GameLift Servers servizio quando è pronto per ospitare sessioni di gioco.

Codice di esempio per GameMode .cpp

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

#include "GameLiftUnrealAppGameMode.h"

#include "UObject/ConstructorHelpers.h"
#include "Kismet/GameplayStatics.h"

#if WITH_GAMELIFT
#include "GameLiftServerSDK.h"
#include "GameLiftServerSDKModels.h"
#endif

#include "GenericPlatform/GenericPlatformOutputDevices.h"

DEFINE_LOG_CATEGORY(GameServerLog);

AGameLiftUnrealAppGameMode::AGameLiftUnrealAppGameMode() :
    ProcessParameters(nullptr)
{
    // Set default pawn class to our Blueprinted character
    static ConstructorHelpers::FClassFinder<APawn> PlayerPawnBPClass(TEXT("/Game/
ThirdPerson/Blueprints/BP_ThirdPersonCharacter"));
```

```
    if (PlayerPawnBPClass.Class != NULL)
    {
        DefaultPawnClass = PlayerPawnBPClass.Class;
    }

    UE_LOG(GameServerLog, Log, TEXT("Initializing AGameLiftUnrealAppGameMode..."));
}

void AGameLiftUnrealAppGameMode::BeginPlay()
{
    Super::BeginPlay();

#if WITH_GAMELIFT
    InitGameLift();
#endif
}

void AGameLiftUnrealAppGameMode::InitGameLift()
{
#if WITH_GAMELIFT
    UE_LOG(GameServerLog, Log, TEXT("Calling InitGameLift..."));

    // Getting the module first.
    FGameLiftServerSDKModule* GameLiftSdkModule =
 &FModuleManager::LoadModuleChecked<FGameLiftServerSDKModule>(FName("GameLiftServerSDK"));

    //Define the server parameters for a GameLift Anywhere fleet. These are not needed
 for a GameLift managed EC2 fleet.
    FServerParameters ServerParametersForAnywhere;

    bool bIsAnywhereActive = false;
    if (FParse::Param(FCommandLine::Get(), TEXT("glAnywhere")))
    {
        bIsAnywhereActive = true;
    }

    if (bIsAnywhereActive)
    {
        UE_LOG(GameServerLog, Log, TEXT("Configuring server parameters for
 Anywhere..."));

        // If GameLift Anywhere is enabled, parse command line arguments and pass them
 in the ServerParameters object.
        FString glAnywhereWebSocketUrl = "";
```

```
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereWebSocketUrl="),
glAnywhereWebSocketUrl))
        {
            ServerParametersForAnywhere.m_webSocketUrl =
TCHAR_TO_UTF8(*glAnywhereWebSocketUrl);
        }

        FString glAnywhereFleetId = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereFleetId="),
glAnywhereFleetId))
        {
            ServerParametersForAnywhere.m_fleetId = TCHAR_TO_UTF8(*glAnywhereFleetId);
        }

        FString glAnywhereProcessId = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereProcessId="),
glAnywhereProcessId))
        {
            ServerParametersForAnywhere.m_processId =
TCHAR_TO_UTF8(*glAnywhereProcessId);
        }
        else
        {
            // If no ProcessId is passed as a command line argument, generate a
randomized unique string.
            FString TimeString = FString::FromInt(std::time(nullptr));
            FString ProcessId = "ProcessId_" + TimeString;
            ServerParametersForAnywhere.m_processId = TCHAR_TO_UTF8(*ProcessId);
        }

        FString glAnywhereHostId = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereHostId="),
glAnywhereHostId))
        {
            ServerParametersForAnywhere.m_hostId = TCHAR_TO_UTF8(*glAnywhereHostId);
        }

        FString glAnywhereAuthToken = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereAuthToken="),
glAnywhereAuthToken))
        {
            ServerParametersForAnywhere.m_authToken =
TCHAR_TO_UTF8(*glAnywhereAuthToken);
        }
```

```
       FString glAnywhereAwsRegion = "";
       if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereAwsRegion="),
glAnywhereAwsRegion))
       {
           ServerParametersForAnywhere.m_awsRegion =
TCHAR_TO_UTF8(*glAnywhereAwsRegion);
       }

       FString glAnywhereAccessKey = "";
       if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereAccessKey="),
glAnywhereAccessKey))
       {
           ServerParametersForAnywhere.m_accessKey =
TCHAR_TO_UTF8(*glAnywhereAccessKey);
       }

       FString glAnywhereSecretKey = "";
       if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereSecretKey="),
glAnywhereSecretKey))
       {
           ServerParametersForAnywhere.m_secretKey =
TCHAR_TO_UTF8(*glAnywhereSecretKey);
       }

       FString glAnywhereSessionToken = "";
       if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereSessionToken="),
glAnywhereSessionToken))
       {
           ServerParametersForAnywhere.m_sessionToken =
TCHAR_TO_UTF8(*glAnywhereSessionToken);
       }

       UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_YELLOW);
       UE_LOG(GameServerLog, Log, TEXT(">>>> WebSocket URL: %s"),
*ServerParametersForAnywhere.m_webSocketUrl);
       UE_LOG(GameServerLog, Log, TEXT(">>>> Fleet ID: %s"),
*ServerParametersForAnywhere.m_fleetId);
       UE_LOG(GameServerLog, Log, TEXT(">>>> Process ID: %s"),
*ServerParametersForAnywhere.m_processId);
       UE_LOG(GameServerLog, Log, TEXT(">>>> Host ID (Compute Name): %s"),
*ServerParametersForAnywhere.m_hostId);
       UE_LOG(GameServerLog, Log, TEXT(">>>> Auth Token: %s"),
*ServerParametersForAnywhere.m_authToken);
```

```
        UE_LOG(GameServerLog, Log, TEXT(">>>> Aws Region: %s"),
 *ServerParametersForAnywhere.m_awsRegion);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Access Key: %s"),
 *ServerParametersForAnywhere.m_accessKey);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Secret Key: %s"),
 *ServerParametersForAnywhere.m_secretKey);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Session Token: %s"),
 *ServerParametersForAnywhere.m_sessionToken);
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }

    UE_LOG(GameServerLog, Log, TEXT("Initializing the GameLift Server..."));

    //InitSDK will establish a local connection with GameLift's agent to enable further
 communication.
    FGameLiftGenericOutcome InitSdkOutcome = GameLiftSdkModule-
>InitSDK(ServerParametersForAnywhere);
    if (InitSdkOutcome.IsSuccess())
    {
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_GREEN);
        UE_LOG(GameServerLog, Log, TEXT("GameLift InitSDK succeeded!"));
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }
    else
    {
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_RED);
        UE_LOG(GameServerLog, Log, TEXT("ERROR: InitSDK failed : ("));
        FGameLiftError GameLiftError = InitSdkOutcome.GetError();
        UE_LOG(GameServerLog, Log, TEXT("ERROR: %s"), *GameLiftError.m_errorMessage);
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
        return;
    }

    ProcessParameters = MakeShared<FProcessParameters>();

    //When a game session is created, Amazon GameLift Servers sends an activation
 request to the game server and passes along the game session object containing game
 properties and other settings.
    //Here is where a game server should take action based on the game session object.
    //Once the game server is ready to receive incoming player connections, it should
 invoke GameLiftServerAPI.ActivateGameSession()
    ProcessParameters->OnStartGameSession.BindLambda([=]
(Aws::GameLift::Server::Model::GameSession InGameSession)
        {
```

```
            FString GameSessionId = FString(InGameSession.GetGameSessionId());
            UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"),
*GameSessionId);
            GameLiftSdkModule->ActivateGameSession();
        });

    //OnProcessTerminate callback. Amazon GameLift Servers will invoke this callback
before shutting down an instance hosting this game server.
    //It gives this game server a chance to save its state, communicate with services,
etc., before being shut down.
    //In this case, we simply tell Amazon GameLift Servers we are indeed going to
shutdown.
    ProcessParameters->OnTerminate.BindLambda([=]()
        {
            UE_LOG(GameServerLog, Log, TEXT("Game Server Process is terminating"));
            GameLiftSdkModule->ProcessEnding();
        });

    //This is the HealthCheck callback.
    //Amazon GameLift Servers will invoke this callback every 60 seconds or so.
    //Here, a game server might want to check the health of dependencies and such.
    //Simply return true if healthy, false otherwise.
    //The game server has 60 seconds to respond with its health status. Amazon GameLift
Servers will default to 'false' if the game server doesn't respond in time.
    //In this case, we're always healthy!
    ProcessParameters->OnHealthCheck.BindLambda([]()
        {
            UE_LOG(GameServerLog, Log, TEXT("Performing Health Check"));
            return true;
        });

    //GameServer.exe -port=7777 LOG=server.mylog
    ProcessParameters->port = FURL::UrlConfig.DefaultPort;
    TArray<FString> CommandLineTokens;
    TArray<FString> CommandLineSwitches;

    FCommandLine::Parse(FCommandLine::Get(), CommandLineTokens, CommandLineSwitches);

    for (FString SwitchStr : CommandLineSwitches)
    {
        FString Key;
        FString Value;

        if (SwitchStr.Split("=", &Key, &Value))
```

```
        {
            if (Key.Equals("port"))
            {
                ProcessParameters->port = FCString::Atoi(*Value);
            }
        }
    }

    //Here, the game server tells Amazon GameLift Servers where to find game session
 log files.
    //At the end of a game session, Amazon GameLift Servers uploads everything in the
 specified
    //location and stores it in the cloud for access later.
    TArray<FString> Logfiles;
    Logfiles.Add(TEXT("GameServerLog/Saved/Logs/GameServerLog.log"));
    ProcessParameters->logParameters = Logfiles;

    //The game server calls ProcessReady() to tell Amazon GameLift Servers it's ready
 to host game sessions.
    UE_LOG(GameServerLog, Log, TEXT("Calling Process Ready..."));
    FGameLiftGenericOutcome ProcessReadyOutcome = GameLiftSdkModule-
>ProcessReady(*ProcessParameters);

    if (ProcessReadyOutcome.IsSuccess())
    {
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_GREEN);
        UE_LOG(GameServerLog, Log, TEXT("Process Ready!"));
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }
    else
    {
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_RED);
        UE_LOG(GameServerLog, Log, TEXT("ERROR: Process Ready Failed!"));
        FGameLiftError ProcessReadyError = ProcessReadyOutcome.GetError();
        UE_LOG(GameServerLog, Log, TEXT("ERROR: %s"),
 *ProcessReadyError.m_errorMessage);
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }

    UE_LOG(GameServerLog, Log, TEXT("InitGameLift completed!"));
#endif
}
```

Fase 3: Ricostruisci il progetto del gioco

- Crea un progetto di gioco per entrambi i seguenti tipi di target: Development Editor e Development Server.

> **Note**
>
> Non è necessario ricostruire la soluzione. Invece, crea il progetto `/Games/` nella cartella della tua app. In caso contrario, Visual Studio ricostruirà l'intero UE5 progetto, operazione che può richiedere fino a un'ora.

Package del tuo server di gioco per l'hosting

Ora che il codice del tuo server di gioco è integrato con la funzionalità SDK del server minima richiesta, sei pronto per creare un pacchetto di build del tuo server di gioco utilizzando Unreal Editor.

Per impacchettare la build del server di gioco

1. Apri il progetto di gioco nell'Unreal Editor.
2. Segui i passaggi dell'Unreal Editor per impacchettare il tuo server di gioco:
   - Scegli la tua piattaforma di destinazione (Windows o Linux).
   - Seleziona l'obiettivo di build del server (*`[your application name]`*`Server`.

   Il processo di imballaggio genera il file eseguibile del server di gioco:*`[your application name]`*`Server.exe`.
3. Prepara la build del tuo server di gioco per la distribuzione su risorse di hosting. La build dovrebbe includere i seguenti file:
   - Il tuo file eseguibile sul server di gioco
   - Per le build di Windows, includi questi file (li trovi nella versione sorgente di Unreal Engine):
     - `VC_redist.x64.exe` (`UnrealEngine\Engine\Source\Programs\PrereqInstaller\Resources\VCRedist\`)
     - `UEPrereqSetup_x64.exe or UE5PrereqSetup_x64.exe` (`UnrealEngine\Engine\Extras\Redist\en-us\`)
   - Tutte le altre dipendenze richieste per il tuo server di gioco.

- Librerie OpenSSL, se necessario. Controlla la versione SDK del server integrata nella build del tuo server di gioco. Con il server SDK for Unreal, versione 5.3.x, non è più necessario includere manualmente le librerie OpenSSL nella build del server di gioco. Per tutte le versioni precedenti, devi individuare e aggiungere i file della libreria. Visualizza la versione più recente dell'SDK del server.

Le librerie OpenSSL sono necessarie per i server di gioco creati con server SDK per Unreal, versioni 5.2.x e precedenti

Devi includere la stessa versione delle librerie OpenSSL utilizzata per impacchettare il server di gioco in Unreal. Queste librerie si trovano nella fonte del motore di gioco. La posizione varia a seconda dell'ambiente di sviluppo:

In Windows:

- `[ENGINE_ROOT_DIR]\Engine\Extras\ThirdPartyNotUE\libimobiledevice\x64\libssl-1_1-x64.dll`
- `[ENGINE_ROOT_DIR]\Engine\Extras\ThirdPartyNotUE\libimobiledevice\x64\libcrypto-1_1-x64.dll`

In Linux:

- `Engine/Source/Thirdparty/OpenSSL/1.1.1`*`n`*`/include/libssl.so.1.1`
- `Engine/Source/Thirdparty/OpenSSL/1.1.1`*`n`*`/include/libcrypto.so.1.1`

Copia le librerie OpenSSL nel pacchetto di build del gioco nella stessa directory del file eseguibile del server di gioco.

Passaggi successivi

Ora che hai preparato una build di server di gioco con le funzionalità minime richieste per l'hostingAmazon GameLift Servers, considera questi potenziali passaggi successivi:

- Implementa il tuo server di gioco integrato per il test e lo sviluppo. Con una flotta Anywhere, puoi configurare il tuo computer locale come risorsa di hosting e utilizzarlo per testare le connessioni del server di gioco e dei client di gioco. Per l'hosting basato sul cloud, distribuisci il tuo server di gioco su una flotta di container gestita EC2 o gestita. Consulta questi argomenti come guida:

  - Configurato per lo sviluppo iterativo con Amazon GameLift Servers Ovunque
  - Amazon GameLift Servers Flotte ovunque
  - Amazon GameLift Servers EC2 flotte gestite
  - Amazon GameLift Serversflotte di container gestite
- Personalizza l'integrazione del server di gioco aggiungendo funzionalità opzionali. Ad esempio, potresti voler aggiungere sessioni con giocatori unici IDs, impostare il matchmaking backfill o gestire l'accesso del server di gioco alle altre tue AWS risorse (come un database o un servizio di archiviazione dei contenuti). Consulta questi argomenti come guida:
  - Aggiungi Amazon GameLift Servers al tuo server di gioco
  - Server C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers
- Personalizza il componente del client di gioco per richiedere sessioni di gioco, ricevere informazioni di connessione e connetterti direttamente a un server di gioco per giocare. Consulta questi argomenti come guida:
  - Integra il tuo client di gioco conAmazon GameLift Servers

Integrazione Amazon GameLift Servers in un progetto Unity

Scopri come integrare l'Amazon GameLift ServersSDK for Unity nei tuoi progetti di gioco per accedere al set completo di funzionalità SDK del server.

> **Suggerimento**
>
> Per una distribuzione più rapida, prova il plug-in Amazon GameLift Servers standalone per Unity. Fornisce flussi di lavoro guidati per l'interfaccia utente per implementare rapidamente il server di gioco con una configurazione minima, in modo da poter provare i componenti del gioco in azione. Consultare Amazon GameLift Serversplugin per Unity (server SDK 5.x).

Risorse aggiuntive:

- Server C# SDK 5.x per -- Azioni Amazon GameLift Servers
- the section called “Ottieni strumenti di sviluppo”

Installa il server SDK for Unity

Scarica l'open source Amazon GameLift Servers per Unity da. GitHub I file readme del repository contengono i prerequisiti e le istruzioni di installazione.

Configura una flotta Amazon GameLift Servers Anywhere per i test

Puoi configurare la tua workstation di sviluppo come flotta di hosting Amazon GameLift Servers Anywhere per testare in modo iterativo la tua Amazon GameLift Servers integrazione. Con questa configurazione, puoi avviare i processi del server di gioco sulla tua workstation, inviare richieste di iscrizione ai giocatori o di matchmaking per Amazon GameLift Servers avviare sessioni di gioco e connettere i client alle nuove sessioni di gioco. Con la tua workstation configurata come server di hosting, puoi monitorare tutti gli aspetti dell'integrazione del gioco con. Amazon GameLift Servers

Per istruzioni sulla configurazione della workstation, consulta Configura i test locali con Amazon GameLift Servers Anywhere i seguenti passaggi:

1. Crea una posizione personalizzata per la tua workstation.
2. Crea una flotta Amazon GameLift Servers Anywhere con la tua nuova posizione personalizzata. In caso di successo, questa richiesta restituisce un ID della flotta. Prendi nota di questo valore, poiché ti servirà in seguito.
3. Registra la tua workstation come computer nella nuova flotta Anywhere. Fornisci un nome di elaborazione univoco e specifica l'indirizzo IP della tua workstation. In caso di successo, questa richiesta restituisce un endpoint SDK del servizio, sotto forma di URL. WebSocket Prendi nota di questo valore, poiché ti servirà in seguito.
4. Genera un token di autenticazione per il calcolo della tua workstation. Questa autenticazione di breve durata include il token e una data di scadenza. Il server di gioco lo utilizza per autenticare la comunicazione con il Amazon GameLift Servers servizio. Archivia l'autenticazione sul computer della tua workstation in modo che i processi del server di gioco in esecuzione possano accedervi.

Aggiungi il codice Amazon GameLift Servers del server al tuo progetto Unity

Il tuo server di gioco comunica con il Amazon GameLift Servers servizio per ricevere istruzioni e segnalare lo stato in corso. A tale scopo, aggiungi il codice del server di gioco che utilizza l'SDK del Amazon GameLift Servers server.

L'esempio di codice fornito illustra gli elementi di integrazione di base richiesti. Usa a `MonoBehavior` per illustrare una semplice inizializzazione del server di gioco con. Amazon GameLift Servers

L'esempio presuppone che il server di gioco funzioni su una flotta Amazon GameLift Servers Anywhere per i test. Include codice per:

- Inizializza un client Amazon GameLift Servers API. L'esempio utilizza la versione `InitSDK()` con parametri del server per la flotta e l'elaborazione di Anywhere. Utilizza l' WebSocket URL, l'ID della flotta, il nome di calcolo (ID host) e il token di autenticazione, come definito nell'argomento precedente. Configura una flotta Amazon GameLift Servers Anywhere per i test
- Implementa le funzioni di callback per rispondere alle richieste del Amazon GameLift Servers servizio, tra cui `OnStartGameSession``OnProcessTerminate`, e. `onHealthCheck`
- Chiama ProcessReady () con una porta designata per avvisare il Amazon GameLift Servers servizio quando il processo è pronto per ospitare sessioni di gioco.

Il codice di esempio fornito stabilisce la comunicazione con il Amazon GameLift Servers servizio. Implementa inoltre una serie di funzioni di callback che rispondono alle richieste del servizio. Amazon GameLift Servers Per ulteriori informazioni su ciascuna funzione e sul funzionamento del codice, consulta Inizializzare il processo del server. Per ulteriori informazioni sulle azioni SDK e sui tipi di dati utilizzati in questo codice, leggi. Server C# SDK 5.x per -- Azioni Amazon GameLift Servers

Il codice di esempio mostra come aggiungere la funzionalità richiesta, come descritto in Aggiungi Amazon GameLift Servers al tuo server di gioco. Per ulteriori informazioni sulle azioni dell'SDK del server, consultaServer C# SDK 5.x per -- Azioni Amazon GameLift Servers.

Codice di esempio di integrazione

```
using System.Collections.Generic;
using Aws.GameLift.Server;
using UnityEngine;

public class ServerSDKManualTest : MonoBehaviour
{
    //This example is a simple integration that initializes a game server process
    //that is running on an Amazon GameLift Servers Anywhere fleet.
    void Start()
    {
        //Identify port number (hard coded here for simplicity) the game server is
 listening on for player connections
        var listeningPort = 7777;

        //WebSocketUrl from RegisterHost call
        var webSocketUrl = "wss://us-west-2.api.amazongamelift.com";
```

```
        //Unique identifier for this process
        var processId = "myProcess";

        //Unique identifier for your host that this process belongs to
        var hostId = "myHost";

        //Unique identifier for your fleet that this host belongs to
        var fleetId = "myFleet";

        //Authorization token for this host process
        var authToken = "myAuthToken";

        //Server parameters are required for an Amazon GameLift Servers Anywhere fleet.
        //They are not required for an Amazon GameLift Servers managed EC2 fleet.
        ServerParameters serverParameters = new ServerParameters(
            webSocketUrl,
            processId,
            hostId,
            fleetId,
            authToken);

        //InitSDK establishes a local connection with an Amazon GameLift Servers agent
        //to enable further communication.
        var initSDKOutcome = GameLiftServerAPI.InitSDK(serverParameters);
        if (initSDKOutcome.Success)
        {
            //Implement callback functions
            ProcessParameters processParameters = new ProcessParameters(
            //Implement OnStartGameSession callback
                (gameSession) => {
                    //Amazon GameLift Servers sends a game session activation request
to the game server
                    //with game session object containing game properties and other
settings.
                    //Here is where a game server takes action based on the game
session object.
                    //When the game server is ready to receive incoming player
connections,
                    //it invokes the server SDK call ActivateGameSession().
                    GameLiftServerAPI.ActivateGameSession();
                },
                (updateGameSession) => {
```

```
                    //Amazon GameLift Servers sends a request when a game session is
updated (such as for
                    //FlexMatch backfill) with an updated game session object.
                    //The game server can examine matchmakerData and handle new
incoming players.
                    //updateReason explains the purpose of the update.
                },
                () => {
                    //Implement callback function OnProcessTerminate
                    //Amazon GameLift Servers invokes this callback before shutting
down the instance hosting this game server.
                    //It gives the game server a chance to save its state, communicate
with services, etc.,
                    //and initiate shut down. When the game server is ready to shut
down, it invokes the
                    //server SDK call ProcessEnding() to tell Amazon GameLift Servers
it is shutting down.
                    GameLiftServerAPI.ProcessEnding();
                },
                () => {
                    //Implement callback function OnHealthCheck
                    //Amazon GameLift Servers invokes this callback approximately every
60 seconds.
                    //A game server might want to check the health of dependencies,
etc.
                    //Then it returns health status true if healthy, false otherwise.
                    //The game server must respond within 60 seconds, or Amazon
GameLift Servers records 'false'.
                    //In this example, the game server always reports healthy.
                    return true;
                },
                //The game server gets ready to report that it is ready to host game
sessions
                //and that it will listen on port 7777 for incoming player connections.
                listeningPort,
                new LogParameters(new List<string>()
                {
                    //Here, the game server tells Amazon GameLift Servers where to find
game session log files.
                    //At the end of a game session, Amazon GameLift Servers uploads
everything in the specified
                    //location and stores it in the cloud for access later.
                    "/local/game/logs/myserver.log"
                }));
```

```
            //The game server calls ProcessReady() to tell Amazon GameLift Servers it's
 ready to host game sessions.
            var processReadyOutcome =
 GameLiftServerAPI.ProcessReady(processParameters);
            if (processReadyOutcome.Success)
            {
                print("ProcessReady success.");
            }
            else
            {
                print("ProcessReady failure : " +
 processReadyOutcome.Error.ToString());
            }
        }
        else
        {
            print("InitSDK failure : " + initSDKOutcome.Error.ToString());
        }
    }

    void OnApplicationQuit()
    {
        //Make sure to call GameLiftServerAPI.ProcessEnding() and
 GameLiftServerAPI.Destroy() before terminating the server process.
        //These actions notify Amazon GameLift Servers that the process is terminating
 and frees the API client from memory.
        GenericOutcome processEndingOutcome = GameLiftServerAPI.ProcessEnding();
        GameLiftServerAPI.Destroy();
        if (processEndingOutcome.Success)
        {
            Environment.Exit(0);
        }
        else
        {
            Console.WriteLine("ProcessEnding() failed. Error: " +
 processEndingOutcome.Error.ToString());
            Environment.Exit(-1);
        }
    }
}
```

Passaggi successivi

Ora che hai preparato una build di server di gioco con le funzionalità minime richieste per l'hostingAmazon GameLift Servers, considera questi potenziali passaggi successivi:

- Implementa il tuo server di gioco integrato per il test e lo sviluppo. Con una flotta Anywhere, puoi configurare il tuo computer locale come risorsa di hosting e utilizzarlo per testare le connessioni del server di gioco e dei client di gioco. Per l'hosting basato sul cloud, distribuisci il tuo server di gioco su una flotta di container gestita EC2 o gestita. Consulta questi argomenti come guida:
  - Configurato per lo sviluppo iterativo con Amazon GameLift Servers Ovunque
  - Amazon GameLift Servers Flotte ovunque
  - Amazon GameLift Servers EC2 flotte gestite
  - Amazon GameLift Serversflotte di container gestite
- Personalizza l'integrazione del server di gioco aggiungendo funzionalità opzionali. Ad esempio, potresti voler aggiungere sessioni con giocatori unici IDs, impostare il matchmaking backfill o gestire l'accesso del server di gioco alle altre tue AWS risorse (come un database o un servizio di archiviazione dei contenuti). Consulta questi argomenti come guida:
  - Aggiungi Amazon GameLift Servers al tuo server di gioco
  - Server C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers
- Personalizza il componente del client di gioco per richiedere sessioni di gioco, ricevere informazioni di connessione e connetterti direttamente a un server di gioco per giocare. Consulta questi argomenti come guida:
  - Integra il tuo client di gioco conAmazon GameLift Servers

Progetta il tuo servizio client di gioco

Ti consigliamo di implementare un servizio client di gioco che autentichi i tuoi giocatori e comunichi con i Amazon GameLift Servers API. Implementando un servizio client di gioco personalizzato, puoi:

- Personalizza l'autenticazione per i tuoi giocatori.
- Controlla come Amazon GameLift Servers abbina e avvia le sessioni di gioco.
- Usa il tuo database di giocatori per gli attributi dei giocatori come la valutazione delle abilità per il matchmaking invece di fidarti del cliente.

L'utilizzo di un servizio client di gioco riduce anche i rischi per la sicurezza introdotti dai client di gioco che interagiscono direttamente con i Amazon GameLift Servers API.

Autenticazione dei giocatori

Puoi utilizzare Amazon Cognito e la sessione giocatore IDs per autenticare i tuoi client di gioco. Per gestire il ciclo di vita e le proprietà delle identità dei giocatori, utilizza i pool di utenti di Amazon Cognito.

Se preferisci, crea una soluzione di identità personalizzata e ospitala su. AWS Puoi anche utilizzare gli autorizzatori Lambda per la logica di autorizzazione personalizzata con API Gateway.

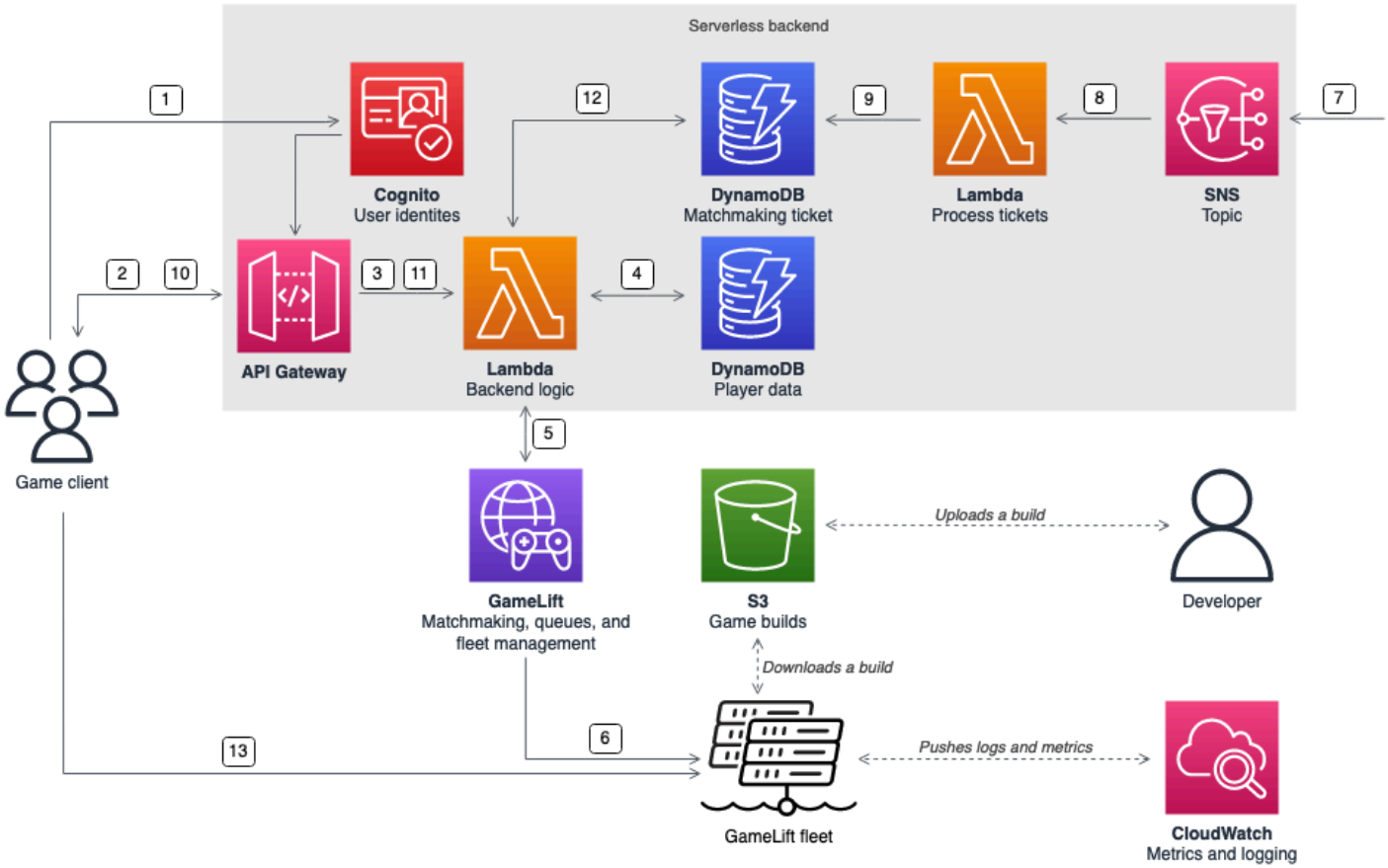
Risorse aggiuntive:

- Utilizzo di pool di identità (identità federate) (Amazon Cognito Developer Guide)
- Guida introduttiva ai pool di utenti (Amazon Cognito Developer Guide)
- Come configurare l'autenticazione dei giocatori con Amazon Cognito (AWS per il blog dei giochi)

Server di sessione di gioco autonomi con un backend senza server

Utilizzando un'architettura di servizio client serverless, il backend può visualizzare lo stato dei ticket di matchmaking da un database altamente scalabile anziché accedendo direttamente a Amazon GameLift Servers API.

Il diagramma seguente mostra un backend serverless costruito con Servizi AWS che abbina i giocatori ai giochi in esecuzione su Amazon GameLift Servers flotte. L'elenco seguente fornisce una descrizione per ogni callout numerato nel diagramma. Per provare questo esempio, vedi Hosting di giochi basato su sessioni multigiocatore on on. AWS GitHub



1. Il client di gioco richiede un'identità utente Amazon Cognito da un pool di identità Amazon Cognito.
2. Il client di gioco riceve credenziali di accesso temporanee e richiede una sessione di gioco tramite un'API Amazon API Gateway.
3. API Gateway richiama una AWS Lambda funzione.
4. La funzione Lambda richiede i dati del giocatore da una tabella NoSQL di Amazon DynamoDB. La funzione fornisce l'identità di Amazon Cognito nei dati del contesto della richiesta.
5. La funzione Lambda richiede una corrispondenza tramite Amazon GameLift Servers FlexMatch matchmaking.
6. FlexMatch abbina un gruppo di giocatori con una latenza adeguata, e quindi richiede il posizionamento di una sessione di gioco tramite un Amazon GameLift Servers coda. La coda ha flotte con una o più Regione AWS postazioni al suo interno.
7. Dopo Amazon GameLift Servers colloca la sessione in una delle sedi della flotta, Amazon GameLift Servers invia una notifica di evento a un argomento di Amazon Simple Notification Service (Amazon SNS).
8. Una funzione Lambda riceve l'evento Amazon SNS e lo elabora.

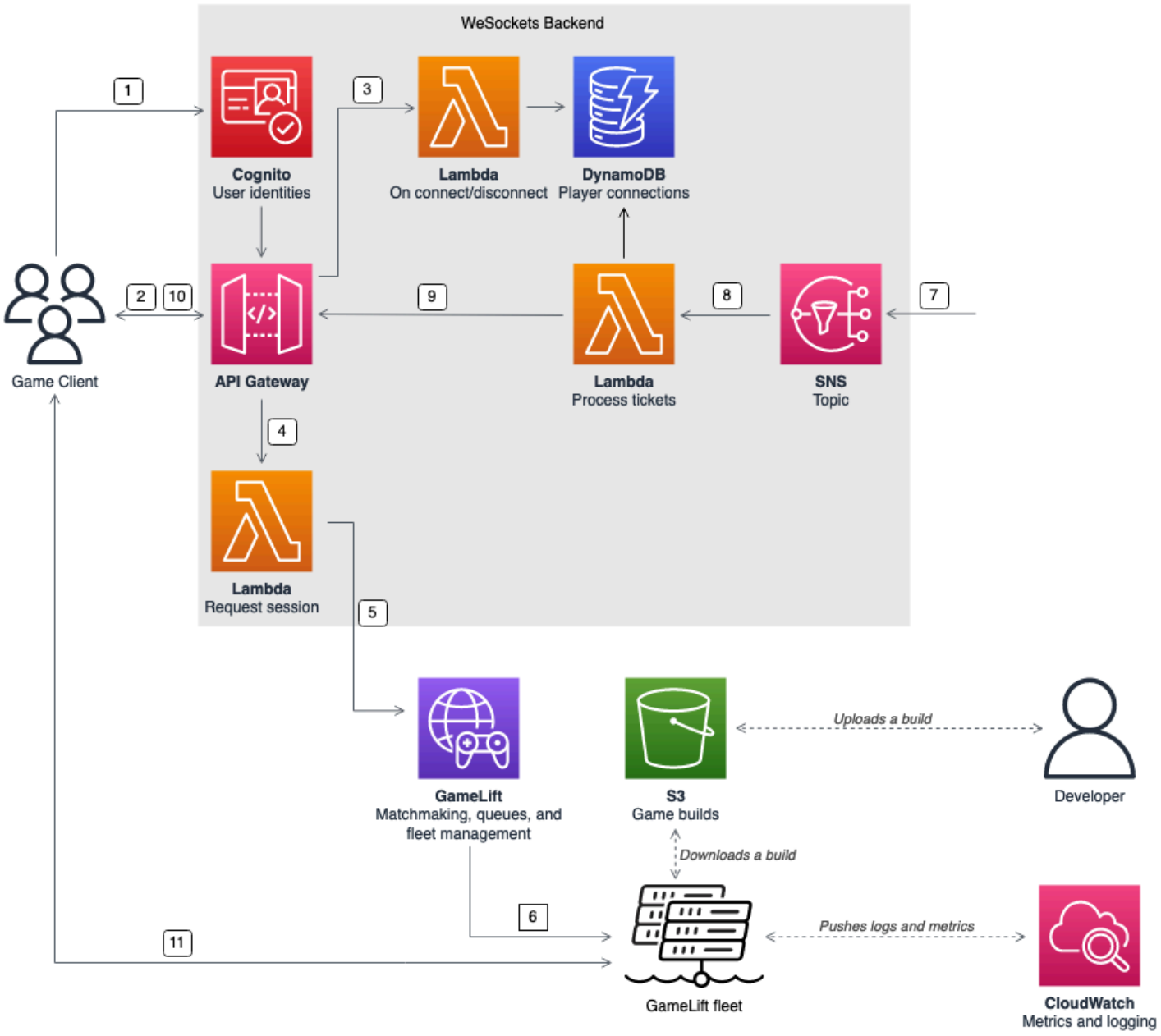
9. Se il ticket di matchmaking è un `MatchmakingSucceeded` evento, la funzione Lambda scrive il risultato, insieme alla porta e all'indirizzo IP del server di gioco, in una tabella DynamoDB.
10. Il client di gioco invia una richiesta firmata ad API Gateway per visualizzare lo stato del ticket di matchmaking in un intervallo specifico.
11. API Gateway utilizza una funzione Lambda che verifica lo stato del ticket di matchmaking.
12. La funzione Lambda controlla la tabella DynamoDB per verificare se il ticket ha esito positivo. Se ha avuto successo, la funzione invia la porta e l'indirizzo IP del server di gioco, insieme all'ID della sessione del giocatore, al client. Se il ticket non è andato a buon fine, la funzione invia una risposta per verificare che la partita non sia ancora pronta.
13. Il client di gioco si connette al server di gioco tramite TCP o UDP utilizzando la porta e l'indirizzo IP forniti dal servizio di backend. Il client di gioco invia quindi l'ID di sessione del giocatore al server di gioco, che quindi convalida l'ID utilizzando l'SDK del server per Amazon GameLift Servers.

Server di sessione di gioco autonomi con un backend basato WebSocket

Utilizzando un'architettura WebSocket basata su Amazon API Gateway, puoi effettuare richieste di matchmaking WebSockets e inviare notifiche push per il completamento del matchmaking utilizzando messaggi avviati dal server. Questa architettura migliora le prestazioni grazie alla comunicazione bidirezionale tra il client e il server.

Per ulteriori informazioni sull'utilizzo di API Gateway WebSock APIs, consulta Working with WebSocket APIs.

Il diagramma seguente mostra un'architettura di backend WebSocket basata su API Gateway e altro Servizi AWS per abbinare i giocatori ai giochi in esecuzione su Amazon GameLift Servers flotte. L'elenco seguente fornisce una descrizione per ogni callout numerato nel diagramma.



1. Il client di gioco richiede un'identità utente Amazon Cognito da un pool di identità Amazon Cognito.
2. Il client di gioco firma una WebSocket connessione a un'API Gateway API con le credenziali di Amazon Cognito.
3. API Gateway richiama una AWS Lambda funzione sulla connessione. La funzione memorizza le informazioni di connessione in una tabella Amazon DynamoDB.
4. Il client di gioco invia un messaggio a una funzione Lambda, tramite l'API API Gateway tramite la WebSocket connessione, per richiedere una sessione.

5. Una funzione Lambda riceve il messaggio e quindi richiede una corrispondenza tramite Amazon GameLift Servers FlexMatch matchmaking.
6. Dopo FlexMatch corrisponde a un gruppo di giocatori, FlexMatch richiede il posizionamento di una sessione di gioco tramite un Amazon GameLift Servers coda.
7. Dopo Amazon GameLift Servers colloca la sessione in una delle sedi della flotta, Amazon GameLift Servers invia una notifica di evento a un argomento di Amazon Simple Notification Service (Amazon SNS).
8. Una funzione Lambda riceve l'evento Amazon SNS e lo elabora.
9. Se il ticket di matchmaking è un `MatchmakingSucceeded` evento, la funzione Lambda richiede la connessione corretta del giocatore da DynamoDB. La funzione invia quindi un messaggio al client di gioco tramite l'API Gateway API tramite la WebSocket connessione. In questa architettura, il client di gioco non verifica attivamente lo stato del matchmaking.
10. Il client di gioco riceve la porta e l'indirizzo IP del server di gioco, insieme all'ID della sessione del giocatore, tramite la WebSocket connessione.
11. Il client di gioco si connette al server di gioco tramite TCP o UDP utilizzando la porta e l'indirizzo IP forniti dal servizio di backend. Il client di gioco invia inoltre l'ID di sessione del giocatore al server di gioco, che quindi convalida l'ID utilizzando l'SDK del server per Amazon GameLift Servers.

Configurato per lo sviluppo iterativo con Amazon GameLift Servers Ovunque

Amazon GameLift Servers fornisce strumenti e soluzioni per aiutarti a configurare un ambiente di test ospitato da utilizzare durante lo sviluppo del gioco. Con questi strumenti, puoi creare un ambiente che rispecchi l'esperienza reale dei giocatori dell'hosting gestito con Amazon GameLift Servers e supporta un processo di sviluppo rapido e iterativo.

Con un ambiente di test separato, si rimuove il sovraccarico di un Amazon GameLift Servers flotta gestita durante i test. Non è più necessario caricare ogni nuova versione di build del server di gioco, creare una nuova flotta e attendere più di 15 minuti prima che venga attivata. Puoi invece creare una nuova build, aggiornare rapidamente la flotta di test con la nuova build, avviarla e iniziare i test.

Usando un Amazon GameLift Servers Ovunque si disponga di un parco macchine, è possibile configurare un ambiente di test utilizzando un dispositivo locale, ad esempio una workstation di sviluppo. Puoi anche configurare un ambiente di test utilizzando una risorsa di hosting basata su cloud.

Configura un ambiente di test Anywhere per sviluppare e testare una serie di scenari, tra cui questi:

- Metti alla prova l'integrazione del tuo server di gioco con Amazon GameLift Servers SDK del server. Puoi eseguire il test anche senza un client di gioco funzionante utilizzando le chiamate AWS CLI per avviare nuove sessioni di gioco e tenere traccia degli eventi delle sessioni di gioco.
- Verifica le interazioni tra il client di gioco, il servizio di backend e il Amazon GameLift Servers assistenza mentre sviluppi componenti per il tuo gioco. Ottimizza l'esperienza del giocatore per partecipare a una partita.
- Sperimenta con il tuo FlexMatch design del matchmaker. Prova le varianti del set di regole e altre implementazioni delle funzionalità di matchmaking. Configura e testa il matchmaking backfill.
- Prova altri Amazon GameLift Servers funzionalità di hosting, come le impostazioni di configurazione del runtime (con Amazon GameLift Servers Agent) per la gestione del ciclo di vita dei server di gioco.
- Crea, testa e ripeti rapidamente per convalidare tutti gli aspetti dell'esperienza di gioco, comprese le interazioni multigiocatore, in un ambiente live ospitato.

In seguito, mentre prepari il gioco per il lancio, ti consigliamo di aggiungerlo Amazon GameLift Servers flotte gestite per ottimizzare le configurazioni di hosting e testare scenari aggiuntivi, tra cui i seguenti:

- Sperimenta e testa i design delle code delle sessioni di gioco, incluso l'uso di flotte con più sedi, flotte Spot e On-Demand e più tipi di istanze.
- Prova le opzioni di posizionamento delle sessioni di gioco con flotte gestite, incluso l'uso di politiche di latenza opzionali e impostazioni di prioritizzazione della flotta.
- Configura la scalabilità della capacità per soddisfare la domanda dei giocatori, utilizzando opzioni di ridimensionamento automatico o manuale.
- Configura con AWS CloudFormation Amazon GameLift Servers flotte gestite per gestire le risorse di hosting a lungo termine.

> **Fast Build Update Tool (solo per lo sviluppo)**
>
> Con EC2 le flotte gestite, per implementare un aggiornamento della build del server di gioco, devi caricare ogni nuova build su Amazon GameLift Servers e crea una nuova flotta dedicata. Il Fast Build Update Tool consente di ignorare questi passaggi durante lo sviluppo, risparmiando tempo e velocizzando l'iterazione dello sviluppo. Con questo strumento, puoi

aggiornare rapidamente i file di build del gioco su tutti i computer di una flotta esistente. Lo strumento offre diverse opzioni: puoi sostituire un'intera build di gioco o modificare 6 file specifici e puoi gestire il riavvio dei processi del server di gioco dopo gli aggiornamenti. Puoi anche usarlo per aggiornare i singoli computer di una flotta.
Per scaricare il Fast Build Update Tool e saperne di più su come usarlo, visita il Amazon GameLift Servers Repo Toolkit per lo strumento di aggiornamento Fast Build in Github.

Argomenti

Crea un ambiente di test basato sul cloud

> **Note**
>
> Questo argomento tratta i test iterativi per i giochi integrati con l'SDK del server per Amazon GameLift Servers versione 5.x. Se il gioco utilizza la versione 4.x o precedente dell'SDK del server, consulta. Testa la tua integrazione utilizzando Amazon GameLift Servers Locale

Usa un Amazon GameLift Servers Anywhere fleet per creare e testare in modo iterativo i componenti del gioco in un ambiente ospitato su cloud. Crea una flotta Anywhere con risorse di hosting e una connessione a Amazon GameLift Servers assistenza, esegui i server di gioco su di essi e verifica le funzionalità di gioco secondo necessità.

Implementa una flotta Anywhere con Amazon GameLift Servers Agente

Se la build del tuo server di gioco è integrata con Amazon GameLift Servers SDK 5.x o versione successiva, puoi distribuirlo su una flotta Anywhere basata su cloud con Amazon GameLift Servers Agente. L'Agent è un processo in background che gestisce i cicli di vita dei server di gioco e altre attività su ogni computer di una flotta. Queste attività includono la registrazione del computer con una flotta Anywhere, l'acquisizione di un token di autenticazione e l'avvio/arresto dei processi del server di gioco in base a una serie di istruzioni. L'agente è controllato dalla configurazione di runtime di una flotta, che puoi aggiornare in qualsiasi momento durante la vita del parco veicoli. (L'agente

viene distribuito automaticamente nelle EC2 flotte gestite). Per ulteriori informazioni e per scaricare l'agente, consulta il Amazon GameLift Servers GitHubarchivio.

Configura test iterativi con Amazon EC2

Utilizza il flusso di lavoro guidato in questo Amazon GameLift Servers soluzione toolkit per configurare un ambiente di hosting basato su cloud che rispecchi l'esperienza di hosting gestito con Amazon GameLift Servers.

Il GitHub repository fornisce un set di script che automatizzano la maggior parte dei processi per la configurazione di un ambiente di test con Amazon GameLift Servers Ovunque e il Amazon GameLift Servers Agente. Fornisce inoltre indicazioni per aggiornare l'ambiente ogni volta che devi testare una nuova build del server di gioco. Puoi eseguire un singolo script che distribuisca un ambiente di test con una build di esempio del server di gioco, oppure puoi seguire ogni passaggio per configurarlo con la build del tuo server di gioco.

In questo flusso di lavoro, lavorerai interamente in AWS Management Console, utilizzando AWS CloudShell per eseguire script e completare attività da riga di comando.

> **Note**
>
> Per le attività di questo tutorial, è necessario un utente con AWS account con le autorizzazioni per i seguenti servizi: Amazon GameLift Servers AWS CloudShell, Amazon S3 EC2, AWS Systems Manager Amazon e. AWS Identity and Access Management Gli utenti con accesso a livello di amministratore all' AWS account dispongono già delle autorizzazioni richieste.

Il flusso di lavoro copre le seguenti attività:

- Package per un server di gioco costruito per Amazon GameLift Servers. Il flusso di lavoro fornisce uno script per creare un server di gioco C++ di esempio, che è già stato integrato con l'SDK del server per Amazon GameLift Servers versione 5.x ed è pronto per l'hosting. In alternativa, puoi lavorare con il tuo progetto di gioco se hai completato l'integrazione.
- Configura un bucket Amazon Simple Storage Service per archiviare build e dipendenze dei server di gioco. Man mano che produci nuove versioni delle tue build di gioco, puoi archiviarle in S3 e utilizzare gli script per aggiornare la flotta di Anywhere per i test di gioco.

- Ottieni e costruisci il Amazon GameLift Servers Agente. L'agente gestisce i processi del server di gioco su una risorsa di hosting in base alla configurazione dell'utente. Utilizza la stessa logica e si comporta in modo identico a Amazon GameLift Servers hosting gestito EC2 .
- Configura una flotta Anywhere per le tue risorse di hosting. Con una flotta Anywhere puoi utilizzare il Amazon GameLift Servers servizio per ospitare risorse non gestite da Amazon GameLift Servers. In questo passaggio, configurerai anche la configurazione di runtime, che indica Amazon GameLift Servers Agente quando e come avviare i processi del server di gioco.
- Configura un' EC2 istanza Amazon. Questo è il tuo ambiente di test per test iterativi. È molto più veloce utilizzare un' EC2 istanza standard anziché un'istanza completamente gestita Amazon GameLift Servers istanza (ottimizzata per l'utilizzo a livello di produzione). Con un' EC2 istanza standard, puoi aggiornare rapidamente e continuamente il server di gioco secondo necessità.
- Implementa la build del tuo server di gioco e Amazon GameLift Servers Agente dell' EC2 istanza Amazon. Il flusso di lavoro fornisce uno script che ottiene la versione più recente della build del gioco e tutte le dipendenze e la EC2 installa sull'istanza. In questo flusso di lavoro, le dipendenze includono Amazon GameLift Servers L'agente e l' CloudWatch agente.
- Avvia il Amazon GameLift Servers Agente. Una volta installato, l'agente si avvia e inizia automaticamente a eseguire le istruzioni. Ciò include:
  - Registra l' EC2 istanza come calcolo nel Amazon GameLift Servers Flotta ovunque.
  - Stabilire una WebSocket connessione con Amazon GameLift Servers servizio e ottieni la configurazione di runtime più recente.
  - Avvia i processi del server di gioco in base alle istruzioni nella configurazione di runtime. In questo flusso di lavoro, all'agente viene richiesto di avviare un singolo processo dell'eseguibile del server di gioco.
- Metti alla prova i tuoi scenari di gioco. Con l'ambiente di test configurato e l'ultima build del server di gioco installata, puoi iniziare i test. Il flusso di lavoro illustra diversi passaggi per il test, incluso l'avvio di una sessione di gioco. Accedi ai log del server di CloudWatch gioco per tenere traccia dei progressi man mano che la sessione di gioco si avvia e si prepara ad accettare giocatori.

  Man mano che sviluppi i componenti del gioco, tra cui un client di gioco e un servizio di backend lato client, puoi includerli nei tuoi scenari di test. Usa un client di gioco per richiedere una sessione di gioco, recupera le informazioni di connessione dal Amazon GameLift Servers servizio, quindi connettiti direttamente alla sessione di gioco.
- Implementa una nuova build del server di gioco e ripeti i test. Man mano che sviluppi il gioco, puoi generare nuove build di server di gioco, quindi distribuirle rapidamente nell'ambiente di test per

i EC2 test. Caricali nel bucket Amazon S3 e utilizza gli script del flusso di lavoro per aggiornare l'ambiente di test.

Trasforma il tuo gioco in Amazon GameLift Servers flotte gestite

Dopo aver completato i test di sviluppo e essere pronto a prepararsi per il lancio, questo è un buon momento per passare a Amazon GameLift Servers flotte gestite. Usa flotte gestite per ottimizzare e testare le tue risorse di hosting di giochi. Implementa la tua soluzione di posizionamento delle sessioni di gioco (code e matchmaker), seleziona l'hardware di hosting (comprese le flotte Spot) e le sedi ottimali e scegli una strategia per aumentare la capacità. Potresti anche iniziare a AWS CloudFormation utilizzarla per gestire in modo più efficiente i cicli di vita di tutte le tue risorse di hosting di giochi, tra cui flotte, code e matchmaker.

La transizione da una flotta di test Anywhere basata sul cloud a una Amazon GameLift Servers flotta gestita. Non è necessario modificare alcun codice di gioco e puoi riutilizzare le stesse code e gli stessi matchmaker. Esegui le seguenti attività:

- Crea un Amazon GameLift Servers crea una risorsa. Con una flotta di test Anywhere, devi distribuire manualmente la build e le dipendenze del tuo server di gioco su ogni computer della flotta. Con una flotta gestita, carica il pacchetto di build del gioco su Amazon GameLift Servers, che lo distribuisce automaticamente su tutti i computer della flotta. Implementa una build di server personalizzata per Amazon GameLift Servers hostingPer ulteriori informazioni su come impacchettare i file di build del gioco e creare una risorsa di compilazione con file in un bucket Amazon S3, consulta la pagina.
- Crea una flotta gestita. Crea una flotta utilizzando la console o la AWS CLI, specificando una EC2 flotta gestita. Questo tipo di parco veicoli richiede impostazioni di configurazione aggiuntive, tra cui la specificazione delle risorse di compilazione e dei tipi di istanze. Puoi utilizzare la stessa configurazione di runtime per gestire il ciclo di vita dei server di gioco su ogni computer della flotta. Vedi Crea una EC2 flotta Amazon GameLift Servers gestita i dettagli sulla creazione di una flotta gestita.
- Reindirizza gli alias del parco veicoli (opzionale). Se configuri alias da utilizzare con le flotte Anywhere, puoi riutilizzare gli stessi alias per le flotte gestite. Vedi Crea un Amazon GameLift Servers alias per i dettagli sulla creazione o l'aggiornamento di un alias.

Configura i test locali con Amazon GameLift Servers Anywhere

> **Note**
>
> Questo argomento tratta dei test locali per i giochi integrati con l'SDK del server per la Amazon GameLift Servers versione 5.x. Se il gioco utilizza la versione 4.x o precedente dell'SDK per server, consulta. Testa la tua integrazione utilizzando Amazon GameLift Servers Locale

Usa una flotta Amazon GameLift Servers Anywhere e il tuo hardware per creare e testare in modo iterativo i componenti del gioco in un ambiente ospitato simulato. Configura una flotta Anywhere e registra un dispositivo locale per stabilire una connessione al Amazon GameLift Servers servizio. Installa il server di gioco integrato sul dispositivo, avvia un processo del server di gioco e verifica le funzionalità di gioco secondo necessità. Puoi aggiornare la build del server di gioco tutte le volte che vuoi per testare ogni nuova build.

Con una flotta Anywhere, puoi eseguire i test utilizzando la AWS CLI o con script di test. Se hai integrato un client di gioco conAmazon GameLift Servers, puoi eseguire il client sullo stesso dispositivo locale o su un dispositivo diverso.

I test locali con una flotta Anywhere sono particolarmente utili per testare l'integrazione del server di gioco conAmazon GameLift Servers. Hai piena visibilità su tutte le attività di hosting sul computer locale, nonché sugli eventi e sui dati di registrazione.

> **Note**
>
> Stai usando il Amazon GameLift Servers plugin per Unreal Engine o Unity? Questi strumenti includono flussi di lavoro guidati per l'impostazione di test locali con una flotta Anywhere. Segui la documentazione per Plugin per Unity: configura i test locali con Anywhere Amazon GameLift Servers oPlugin per Unreal: ospita il tuo gioco localmente con Anywhere Amazon GameLift Servers.

Argomenti

- Esegui un'iterazione sul tuo server di gioco
- Trasforma il tuo gioco in flotte Amazon GameLift Servers gestite

Configura una flotta Anywhere locale

Segui questi passaggi per creare una flotta Anywhere per la tua workstation locale. Per istruzioni dettagliate sull'utilizzo della AWS CLI o del moduloAmazon GameLift Servers, AWS Management Console consulta. Crea un Amazon GameLift Servers Flotta ovunque

Per creare la flotta Anywhere

1. Crea una posizione personalizzata per la tua workstation locale. (AWS CLI o console). Una posizione personalizzata è semplicemente un'etichetta per la risorsa di elaborazione che intendi includere nella tua flotta Anywhere. I nomi delle località personalizzate devono iniziare con`custom-`. Ad esempio: `custom-my_laptop`. Consultare Creare una posizione personalizzata.
2. Crea una flotta Anywhere (AWS CLI o console). In questo passaggio, crea la risorsa del parco veicoli con la posizione personalizzata per la workstation locale. Consultare Crea una flotta Anywhere.

   Prendi nota dell'ID o del valore ARN del nuovo parco veicoli. Tale valore servirà nella fase successiva.
3. Registra la tua workstation locale come unità di calcolo della flotta (solo AWS CLI). Una flotta Anywhere deve disporre di almeno una risorsa di elaborazione per ospitare i server di gioco. Consultare Aggiungi un computer alla flotta. Per aggiungere un computer alla flotta, sono necessarie le seguenti informazioni:

   - Un nome di calcolo. Ogni calcolo di un parco dati deve avere un nome univoco.
   - L'identificatore della flotta Anywhere. Puoi usare `FleetID` o`FleetArn`.
   - Le informazioni sulla connessione del computer. Specifica un `IpAddress` oppure il valore `DnsName`. Ecco come i client Amazon GameLift Servers di gioco si connetteranno ai server di gioco.
   - Una posizione personalizzata nella flotta Anywhere.

   Prendi nota del valore `GameLiftServiceSdkEndpoint` restituito. Questo valore ti servirà quando aggiorni il tuo server di gioco per farlo funzionare su una flotta Anywhere.

Aggiorna e installa il tuo server di gioco

Questa attività presuppone che tu abbia già integrato una build di server di gioco con il Amazon GameLift Servers server SDK 5.x. Il processo di integrazione prevede l'aggiunta di codice al server di gioco in modo che possa interagire con il Amazon GameLift Servers servizio per avviare e gestire le sessioni di gioco.

Per una flotta Anywhere, devi configurare manualmente alcune impostazioni del server di gioco. In una flotta Amazon GameLift Servers gestita, queste impostazioni vengono configurate automaticamente.

Per preparare il tuo server di gioco per una flotta Anywhere

1. Ottieni un token di autenticazione. Il tuo server di gioco deve includere un token di autenticazione in ogni comunicazione con il Amazon GameLift Servers servizio. Amazon GameLift Serversi token di autenticazione sono di breve durata e devono essere aggiornati regolarmente.

   È consigliabile creare uno script per completare le seguenti attività:

   - Richiama l'azione AWS CLI. `get-compute-auth-token`
   - Memorizza il valore del token restituito dove i processi del server di gioco possono recuperarlo, ad esempio in una variabile di ambiente sul calcolo locale.

   Installa lo script con il tuo server di gioco sul computer. Imposta lo script in modo che venga eseguito prima di iniziare il primo processo del server di gioco. Mentre i processi del server di gioco sono attivi, esegui lo script regolarmente per mantenere un token di autenticazione valido. Tutti i processi del server di gioco sul computer possono utilizzare lo stesso token di autenticazione.
2. Aggiorna il codice del server Amazon GameLift Servers di gioco. Quando hai integrato il codice del server di gioco con l'SDK del server perAmazon GameLift Servers, hai aggiunto un invito all'azione`InitSdk()`. Quando il server di gioco funziona su una flotta Anywhere, questa chiamata richiede parametri server aggiuntivi. Per ulteriori informazioni, consulta Inizializza il processo del server la sezione Server SDK 5.x per Amazon GameLift Servers relativa al linguaggio di sviluppo in uso. I parametri del server sono:

   - `webSocketUrl`— Imposta questo parametro sul `GameLiftServiceSdkEndpoint` valore che viene restituito quando registri un calcolo nella flotta.

- `hostId`— Imposta questo parametro sul nome del calcolo che specifichi quando registri un calcolo nella flotta Anywhere.
- `fleetId`— Imposta questo parametro sull'ID della flotta Anywhere.
- `authToken`— Imposta questo parametro sul token che viene restituito in risposta a una richiesta di recupero di un token di autenticazione per un calcolo.
- `processId`— Imposta questo parametro per identificare un processo del server di gioco in esecuzione sul computer locale. Ogni processo simultaneo del server di gioco deve avere un ID di processo univoco.

I valori dei parametri del server utilizzati da ogni processo del server di gioco devono essere specifici per l'elaborazione della flotta Anywhere su cui è in esecuzione il processo. Per i dettagli su come ottenere i valori appropriati per un calcolo, consulta. Aggiungi un computer alla flotta Come procedura consigliata `webSocketUrl` `hostIdfleetId`, imposta e `authToken` come variabili di ambiente nell'elaborazione locale. Tutti i processi del server eseguiti sul computer utilizzeranno questi valori.

3. Installa la build del server di gioco sul computer locale. Includi tutte le dipendenze necessarie per far funzionare il server di gioco.
4. Avvia uno o più processi del server di gioco in esecuzione sul computer locale. Quando il processo del server di gioco richiama l'azione SDK del server`ProcessReady()`, il processo è pronto per ospitare una sessione di gioco.

Verifica l'attività della sessione di gioco

Verifica l'integrazione del tuo server di gioco lavorando con le sessioni di gioco. Se non disponi di un client di gioco integrato con Amazon GameLift Servers funzionalità, puoi utilizzare la AWS CLI per avviare sessioni di gioco. Prova i seguenti scenari:

- Creare una sessione di gioco. create-game-session Comando di chiamata (o operazione CreateGameSessionAPI). Specificate l'ID e la posizione personalizzata della vostra flotta Anywhere. Questa chiamata restituisce un identificatore univoco per la nuova sessione di gioco.
- Controlla lo stato della sessione di gioco.describe-game-sessionsComando di chiamata (o azione DescribeGameSessionsAPI). Specificate l'ID della sessione di gioco. Questa chiamata restituisce informazioni dettagliate sulla sessione di gioco, incluso lo stato della sessione di gioco. Le sessioni di gioco in stato Attivo sono pronte per consentire ai giocatori di connettersi. Per ottenere un

elenco di tutte le sessioni di gioco per la flotta, chiama il list-game-sessionscomando (o l'azione ListGameSessionsAPI).

- Connect alla sessione di gioco. Se il tuo client di gioco ha la possibilità di partecipare a una sessione di gioco, utilizza le informazioni di connessione incluse nelle informazioni sulla sessione di gioco.

Esegui un'iterazione sul tuo server di gioco

Puoi usare la stessa Anywhere flotta e lo stesso calcolo per testare altre versioni della build del tuo server di gioco.

1. Pulisci il tuo esistente**`GameSession`**. Se il processo del server di gioco si blocca o non chiama`ProcessEnding()`, Amazon GameLift Servers pulisce `GameSession` dopo che il server di gioco ha interrotto l'invio dei controlli di integrità.
2. Genera una nuova build del server di gioco. Apporta modifiche al tuo server di gioco e crea una build rivista.
3. Aggiorna la build del server di gioco sul tuo computer locale. La tua flotta Anywhere precedente è ancora attiva e il tuo laptop è ancora registrato come risorsa di elaborazione nel parco.
4. Ottieni un token di autorizzazione aggiornato. Chiama il comando get-compute-auth-tokenCLI e archivia il token sul calcolo locale.
5. Avvia uno o più processi del server di gioco in esecuzione sul computer locale. Quando il processo del server di gioco chiama`ProcessReady()`, è pronto per essere utilizzato per i test.

Trasforma il tuo gioco in flotte Amazon GameLift Servers gestite

Dopo aver completato i test di sviluppo e pronto a prepararti per il lancio, questo è un buon momento per passare alle flotte Amazon GameLift Servers gestite. Usa le flotte gestite per perfezionare e testare le tue risorse di hosting di giochi. Implementa la tua soluzione di posizionamento delle sessioni di gioco (code e matchmaker), seleziona l'hardware di hosting (comprese le flotte Spot) e le sedi ottimali e scegli una strategia per aumentare la capacità. Potresti anche iniziare a AWS CloudFormation utilizzarla per gestire in modo più efficiente i cicli di vita di tutte le tue risorse di hosting di giochi, tra cui flotte, code e matchmaker.

È necessario apportare alcune modifiche minori per passare da una flotta di test Anywhere locale a una flotta gestita. Amazon GameLift Servers Puoi riutilizzare le stesse code e gli stessi matchmaker. Esegui le seguenti attività:

- Cambia la chiamata in codice del server di gioco in**`InitSdk()`**. Rimuovi i parametri del server. Per una flotta gestita, tiene traccia Amazon GameLift Servers automaticamente di queste informazioni.
- Crea una risorsa di Amazon GameLift Servers compilazione. Con una flotta di test Anywhere, devi distribuire manualmente la build e le dipendenze del server di gioco su ogni computer della flotta. Con una flotta gestita, puoi creare e caricare il tuo pacchetto di build di gioco suAmazon GameLift Servers, che lo distribuisce automaticamente su tutti i computer della flotta. Implementa una build di server personalizzata per Amazon GameLift Servers hostingPer ulteriori informazioni su come impacchettare i file di build del gioco e creare una risorsa di compilazione con file in un bucket Amazon S3, consulta la pagina. Non includere script che registrano un'elaborazione e ottengono un token di autenticazione, poiché gestisce Amazon GameLift Servers automaticamente queste attività con flotte gestite.
- Crea una flotta gestita. Crea una flotta utilizzando la console o la AWS CLI, specificando una EC2 flotta gestita. Questo tipo di parco veicoli richiede impostazioni di configurazione aggiuntive, tra cui la specificazione delle risorse di compilazione e dei tipi di istanze. Dovete tutti impostare una configurazione di runtime per gestire il ciclo di vita dei server di gioco su ogni computer della flotta. Crea una EC2 flotta Amazon GameLift Servers gestitaPer ulteriori informazioni sulla creazione di una flotta gestita, consulta la sezione.
- Reindirizza gli alias del parco veicoli (opzionale). Se configuri alias da utilizzare con le tue flotte Anywhere, puoi riutilizzare gli stessi alias per le flotte gestite. Vedi Crea un Amazon GameLift Servers alias per i dettagli sulla creazione o l'aggiornamento di un alias.

Collabora con l'Amazon GameLift Serversagente

L'Amazon GameLift Serversagente supervisiona l'esecuzione dei processi dei server di gioco sulle vostre Amazon GameLift Servers flotte. L'agente viene distribuito su ogni computer di una flotta e fornisce la gestione automatizzata dei processi, la gestione dell'hosting e la registrazione per l'elaborazione. Per utilizzare l'Agent, è necessario che la build del server di gioco sia integrata con l'SDK del server per Amazon GameLift Servers la versione 5.x o successiva.

L'Amazon GameLift Serversagente è disponibile esternamente per l'uso con Amazon GameLift Servers flotte che non sono flotte gestite. EC2 (EC2 Le flotte gestite gestiscono automaticamente le attività dell'agente.) Puoi scegliere di gestire Amazon GameLift Servers flotte, incluse le flotte Anywhere, con o senza l'agente. Senza l'agente, è necessario fornire una soluzione alternativa per il completamento delle attività richieste.

Quando viene distribuito su un computer, l'Amazon GameLift Serversagente deve essere avviato prima dell'avvio di qualsiasi processo del server di gioco. All'avvio, l'agente completa le seguenti attività:

- Registra l'elaborazione con una flotta Amazon GameLift Servers Anywhere utilizzando l'API. RegisterCompute
- Richiama l'GetComputeAuthTokenAPI per recuperare un token di autorizzazione e lo archivia per l'utilizzo da parte dei processi del server in esecuzione sul computer.
- Imposta la variabile di ambiente WebSocket URL per il calcolo e stabilisce una WebSocket connessione al servizio. Amazon GameLift Servers
- Richiede al servizio la versione più recente della configurazione di runtime della Amazon GameLift Servers flotta.
- Avvia e arresta i processi del server in base alle istruzioni di configurazione del runtime.

Il codice sorgente e le istruzioni di compilazione per l'Amazon GameLift Serversagente sono disponibili nell'Amazon GameLift Serversagente GitHub.

Informazioni sull'agente

L'Amazon GameLift Serversagente è progettato per gestire le seguenti attività per le vostre flotte:

Gestione dei processi

- Avvia nuovi processi del server come definito nelle istruzioni di runtime. L'agente potrebbe utilizzare una configurazione di runtime personalizzata distribuita con l'agente. In alternativa, puoi fornire una `RuntimeConfiguration` come parte della definizione del tuo parco veicoli. Questo approccio ha il vantaggio di poter modificare la configurazione di runtime della flotta in qualsiasi momento. L'agente richiede periodicamente configurazioni di runtime aggiornate dal Amazon GameLift Servers servizio.
- Monitora le attivazioni dei processi del server e termina i processi quando non si attivano in tempo.
- Invia battiti cardiaci a. Amazon GameLift Servers Se l'agente non riesce a inviare battiti cardiaci, il computer potrebbe essere contrassegnato come obsoleto.
- Indica Amazon GameLift Servers quando termina un processo del server. Amazon GameLift Serversutilizza queste informazioni per monitorare la disponibilità del server di gioco per il posizionamento delle sessioni di gioco.
- Emette eventi di flotta per i processi del server, tra cui:

- `SERVER_PROCESS_INVALID_PATH`: I parametri di avvio del processo del server di gioco non erano configurati correttamente.
- `SERVER_PROCESS_TERMINATED_UNHEALTHY`: Il processo del server di gioco non ha segnalato un controllo sanitario valido entro 3 minuti dall'attivazione ed è stato quindi interrotto.
- `SERVER_PROCESS_FORCE_TERMINATED`: il processo del server di gioco non è terminato correttamente dopo l'`OnProcessTerminate()`invio entro 30 secondi.
- `SERVER_PROCESS_CRASHED`: Un processo del server di gioco si è bloccato per qualche motivo.

Gestione del calcolo

- Riceve messaggi dal Amazon GameLift Servers servizio per arrestare l'elaborazione.
- Richiede l'interruzione del calcolo da. Amazon GameLift Servers

Registrazione di log

- Carica i log in un bucket Amazon S3 nel tuo account. AWS

Testa la tua integrazione utilizzando Amazon GameLift Servers Locale

> **Note**
>
> Questo argomento tratta dei test per i giochi integrati con l'SDK del server per Amazon GameLift Servers solo versione 4.x o precedente. Il pacchetto SDK del server include una versione compatibile di Amazon GameLift Servers Locale. Se utilizzi la versione 5.x dell'SDK del server, consulta Configura i test locali con Amazon GameLift Servers Anywhere per i test locali con un Amazon GameLift Servers Flotta ovunque.

Utilizzo Amazon GameLift Servers Locale per eseguire una versione limitata del file gestito Amazon GameLift Servers esegui il servizio su un dispositivo locale e verifica l'integrazione del gioco su di esso. Questo strumento è utile quando si esegue lo sviluppo iterativo sull'integrazione dei giochi. L'alternativa: caricare ogni nuova build su Amazon GameLift Servers e la configurazione di una flotta per ospitare la partita può richiedere diverse o più operazioni ogni volta.

Con Amazon GameLift Servers A livello locale, puoi verificare quanto segue:

- Il tuo server di gioco è correttamente integrato con Server SDK e comunica correttamente con il Amazon GameLift Servers servizio per avviare nuove sessioni di gioco, accettare nuovi giocatori e segnalare lo stato e lo stato di salute.
- Il tuo client di gioco è correttamente integrato con l' AWS SDK per Amazon GameLift Servers ed è in grado di recuperare informazioni sulle sessioni di gioco esistenti, avviare nuove sessioni di gioco, unire giocatori alle partite e connettersi alla sessione di gioco.

Amazon GameLift Servers Local è uno strumento da riga di comando che avvia una versione autonoma del file gestito Amazon GameLift Servers servizio. Amazon GameLift Servers Local fornisce anche un registro degli eventi in esecuzione relativo all'inizializzazione dei processi del server, ai controlli di integrità e alle chiamate e risposte delle API. Amazon GameLift Servers Local riconosce un sottoinsieme delle azioni SDK per AWS Amazon GameLift Servers. Puoi effettuare chiamate dal AWS CLI o dal tuo client di gioco. Tutte le azioni API vengono eseguite localmente proprio come in Amazon GameLift Servers servizio web.

Ogni processo del server dovrebbe ospitare solo una singola sessione di gioco. La sessione di gioco è l'eseguibile a cui ti connetti Amazon GameLift Servers Locale. Quando la sessione di gioco è completata, dovresti chiamare `GameLiftServerSDK::ProcessEnding` e poi uscire dal processo. Durante il test in locale con Amazon GameLift Servers In locale, è possibile avviare più processi server. Ogni processo si connetterà a Amazon GameLift Servers Locale. Puoi quindi creare una sessione di gioco per ogni processo del server. Al termine della sessione di gioco, il processo del server di gioco dovrebbe chiudersi. È quindi necessario avviare manualmente un altro processo del server.

Amazon GameLift Servers local supporta quanto segue APIs:

- CreateGameSession
- CreatePlayerSession
- CreatePlayerSessions
- DescribeGameSessions
- DescribePlayerSessions

Configurazione Amazon GameLift Servers local

Amazon GameLift Servers Local viene fornito come `.jar` file eseguibile fornito in bundle con Server SDK. Può essere eseguito su Windows o Linux e utilizzato con qualsiasi Amazon GameLift Servers-lingua supportata.

Prima di eseguire Local, è necessaria l'installazione di quanto segue.

- Una build del server SDK per Amazon GameLift Servers versione da 3.1.5 a 4.x.
- Java 8

Prova un server di gioco

Se desideri testare solo il tuo server di gioco, puoi utilizzare il AWS CLI per simulare le chiamate dei client di gioco al Amazon GameLift Servers Servizio locale. In questo modo si verifica che il server di gioco funziona come previsto nelle condizioni seguenti:

- Il server di gioco si avvia correttamente e inizializza l'SDK del server per Amazon GameLift Servers.
- Come parte del processo di avvio, il server di gioco invia una notifica Amazon GameLift Servers che il server è pronto per ospitare sessioni di gioco.
- Il server di gioco invia lo stato di salute a Amazon GameLift Servers ogni minuto durante la corsa.
- Il server di gioco risponde alle richieste di avvio di una nuova sessione di gioco.

1. Inizia Amazon GameLift Servers Locale.

   Aprire una finestra del prompt dei comandi, accedere alla directory contente il file *`GameLiftLocal.jar`* ed eseguire il file. Per impostazione predefinita, Local riceve le richieste provenienti dai client di gioco sulla porta 8080. Per specificare un numero di porta diverso, utilizzare il parametro `-p`, come indicato nell'esempio seguente:

   ```
   java -jar GameLiftLocal.jar -p 9080
   ```

   Una volta avviato Local, è possibile visualizzare i log che indicano che sono stati avviati due server locali, di cui uno rileva il server di gioco e l'altro il client di gioco o la AWS CLI. I log continuano a fornire rapporti sull'attività dei due server locali, comprese le comunicazioni da e verso i componenti di gioco.

2. Avviare il server di gioco.

   Inizia il tuo Amazon GameLift Servers-server di gioco integrato localmente. Non è necessario modificare l'endpoint per il server di gioco.

   Nella finestra del prompt dei comandi locale, i messaggi di registro indicano che il server di gioco si è connesso al Amazon GameLift Servers Servizio locale. Ciò significa che il server di gioco ha inizializzato con successo l'SDK del server per Amazon GameLift Servers (con`InitSDK()`). Ha chiamato `ProcessReady()` con i percorsi di log mostrati e, in caso di esito positivo, è pronto per l'hosting di una sessione di gioco. Mentre il server di gioco è in esecuzione, Amazon GameLift Servers registra ogni rapporto sullo stato di salute del server di gioco. Il seguente esempio di messaggistica di log mostra un server di gioco integrato:

```
16:50:53,217  INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - SDK
 connected: /127.0.0.1:64247
16:50:53,217  INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - SDK pid is 17040,
 sdkVersion is 3.1.5 and sdkLanguage is CSharp
16:50:53,217  INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - NOTE: Only SDK
 versions 3.1.5 and above are supported in GameLiftLocal!
16:50:53,451  INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - onProcessReady
 received from: /127.0.0.1:64247 and ackRequest requested? true
16:50:53,543  INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - onProcessReady
 data: logPathsToUpload: "C:\\game\\logs"
logPathsToUpload: "C:\\game\\error"
port: 1935

16:50:53,544  INFO || - [HostProcessManager] nioEventLoopGroup-3-1 - Registered new
 process true, true,
16:50:53,558  INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - onReportHealth
 received from /127.0.0.1:64247 with health status: healthy
```

   Potenziali messaggi di errore e avviso sono i seguenti:

   - Errore: "ProcessReady non ho trovato un processo con PID:*<process ID>*! È stato invocato initSDK ()?»
   - Avviso: «Lo stato del processo esiste già per il processo con PID:*<process ID>*! ProcessReady(...) è stato invocato più di una volta?»

3. Avvia il. AWS CLI

Una volta che il server di gioco ha completato la chiamata a `ProcessReady()`, è possibile iniziare a effettuare chiamate client. Aprire un'altra finestra del prompt dei comandi e avviare lo strumento AWS CLI . Per AWS CLI impostazione predefinita, utilizza il Amazon GameLift Servers endpoint del servizio web. È necessario sostituirlo con l'endpoint di Local in ogni richiesta con il parametro `--endpoint-url`, come illustrato nella seguente richiesta di esempio.

```
AWS gamelift describe-game-sessions --endpoint-url http://localhost:9080  --fleet-
id fleet-123
```

Nella finestra del AWS CLI prompt dei `AWS gamelift` comandi, i comandi generano risposte come documentato nel Command Reference.AWS CLI

4. Creare una sessione di gioco.

   Con AWS CLI, invia una richiesta CreateGameSession(). La richiesta deve seguire la sintassi prevista. Per Local, il parametro `FleetId` può essere impostato su qualsiasi stringa valida (`^fleet-\S+`).

   ```
   AWS gamelift create-game-session --endpoint-url http://localhost:9080 --maximum-
   player-session-count 2 --fleet-id
       fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d
   ```

   Nella finestra del prompt dei comandi locale, i messaggi di registro indicano che Amazon GameLift Servers Local ha inviato una on`StartGameSession` richiamata al server di gioco. Se una sessione di gioco è creata, il server di gioco risponde richiamando `ActivateGameSession`.

   ```
   13:57:36,129  INFO || - [SDKInvokerImpl]
           Thread-2 - Finished sending event to game server to start a game session:
           arn:aws:gamelift:local::gamesession/
   fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-ab423a4b-b827-4765-
   aea2-54b3fa0818b6.
           Waiting for ack response.13:57:36,143  INFO || - [SDKInvokerImpl]
           Thread-2 - Received ack response: true13:57:36,144  INFO || -
           [CreateGameSessionDispatcher] Thread-2 - GameSession with id:
           arn:aws:gamelift:local::gamesession/
   fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-ab423a4b-b827-4765-
   aea2-54b3fa0818b6
   ```

```
        created13:57:36,227  INFO || - [SDKListenerImpl]
        nioEventLoopGroup-3-1 - onGameSessionActivate received
 from: /127.0.0.1:60020 and ackRequest
        requested? true13:57:36,230  INFO || - [SDKListenerImpl]
        nioEventLoopGroup-3-1 - onGameSessionActivate data: gameSessionId:
        "arn:aws:gamelift:local::gamesession/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-abcdef12-3456-7890-abcd-
ef1234567890"
```

Nella AWS CLI finestra, Amazon GameLift Servers risponde con un oggetto della sessione di gioco che include un ID della sessione di gioco. Si noti che lo stato della nuova sessione di gioco è Activating (In fase di attivazione). Lo stato diventa Attivo una volta ActivateGameSession richiamato il server di gioco. Se vuoi vedere lo stato modificato, usa il comando AWS CLI per chiamare. `DescribeGameSessions()`

```
{
    "GameSession": {
      "Status": "ACTIVATING",
      "MaximumPlayerSessionCount": 2,
      "FleetId": "fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d",
      "GameSessionId": "arn:aws:gamelift:local::gamesession/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-abcdef12-3456-7890-abcd-
ef1234567890",
      "IpAddress": "127.0.0.1",
      "Port": 1935
    }
}
```

Prova un server e un client di gioco

Per verificare l'integrazione completa dei giochi, compresa la connessione dei giocatori ai giochi, è possibile eseguire sia il server sia il client di gioco a livello locale. Ciò ti consente di testare le chiamate programmatiche dal tuo client di gioco al Amazon GameLift Servers Locale. È possibile verificare le seguenti azioni:

- Il client di gioco sta effettuando correttamente le richieste AWS SDK a Amazon GameLift Servers Servizio locale, che include la creazione di sessioni di gioco, il recupero di informazioni sulle sessioni di gioco esistenti e la creazione di sessioni di gioco.

- Il server di gioco convalida correttamente i giocatori quando tentano di partecipare a una sessione di gioco. Per i giocatori convalidati, il server di gioco potrebbe recuperare i dati giocatore (se la funzione è implementata).
- Il server di gioco segnala l'interruzione di una connessione quando un giocatore esce dal gioco.
- Il server di gioco segnala la fine di una sessione di gioco.

1. Inizia Amazon GameLift Servers Locale.

   Aprire una finestra del prompt dei comandi, accedere alla directory contente il file *`GameLiftLocal.jar`* ed eseguire il file. Per impostazione predefinita, Local riceve le richieste provenienti dai client di gioco sulla porta 8080. Per specificare un numero di porta diverso, utilizzare il parametro `-p`, come indicato nell'esempio seguente.

   ```
   ./gamelift-local -p 9080
   ```

   Una volta avviato Local, è possibile visualizzare i log che indicano che sono stati avviati due server locali, di cui uno rileva il server di gioco e l'altro il client di gioco o la AWS CLI.

2. Avviare il server di gioco.

   Inizia il tuo Amazon GameLift Servers-server di gioco integrato localmente. Per ulteriori informazioni sui log dei messaggi, consultare Prova un server di gioco.

3. Configurare il client di gioco per Local e avviarlo.

   Per utilizzare il client di gioco con Amazon GameLift Servers Servizio locale, devi apportare le seguenti modifiche alla configurazione del client di gioco, come descritto inConfigura Amazon GameLift Servers su un servizio di backend:

   - Modificare l'oggetto `ClientConfiguration` in modo che punti all'endpoint Local, ad esempio `http://localhost:9080`.
   - Impostare un valore di ID del parco istanze di destinazione. Per Local, non è necessario un vero e proprio ID del parco istanze; impostare il parco istanze di destinazione su qualsiasi stringa valida (`^fleet-\S+`), ad esempio `fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d`.
   - Imposta AWS le credenziali. Per Local, non occorrono vere e proprie credenziali AWS ; è possibile impostare la chiave di accesso e la chiave segreta su qualsiasi stringa.

Nella finestra del prompt dei comandi locale, una volta avviato il client di gioco, i messaggi di registro dovrebbero indicare che il client di gioco è stato inizializzato `GameLiftClient` e che la comunicazione è avvenuta con successo con il Amazon GameLift Servers servizio.

4. Prova le chiamate dei client di gioco a Amazon GameLift Servers servizio.

   Verificare che il client di gioco esegua una o tutte le seguenti chiamate API:

   - CreateGameSession()
   - DescribeGameSessions()
   - CreatePlayerSession()
   - CreatePlayerSessions()
   - DescribePlayerSessions()

   Nella finestra del prompt dei comandi di Local, solo le chiamate a `CreateGameSession()` determinano messaggi di log. I messaggi di registro mostrano quando Amazon GameLift Servers Local richiede al server di gioco di avviare una sessione di gioco (`onStartGameSession`callback) e ha esito positivo `ActivateGameSession` quando il server di gioco la richiama. Nella finestra della AWS CLI , tutte le chiamate API determinano risposte o messaggi di errore come documentato.

5. Verificare che il server di gioco convalidi le nuove connessioni dei giocatori.

   Dopo aver creato una sessione di gioco e una sessione giocatore, stabilire una connessione diretta alla sessione di gioco.

   Nella finestra del prompt dei comandi di Local, i messaggi di log devono mostrare che il server di gioco ha inviato una richiesta `AcceptPlayerSession()` per convalidare la nuova connessione dei giocatori. Se usi il comando AWS CLI to call`DescribePlayerSessions()`, lo stato della sessione del giocatore dovrebbe cambiare da Riservato ad Attivo.

6. Verifica che il server di gioco stia segnalando lo stato del gioco e del giocatore al Amazon GameLift Servers servizio.

   In Amazon GameLift Servers per gestire la domanda dei giocatori e riportare correttamente le metriche, il tuo server di gioco deve riportare vari stati a Amazon GameLift Servers. Verifica che Local stia registrando gli eventi relativi alle seguenti azioni. Potresti anche voler utilizzare il per tenere traccia delle AWS CLI modifiche allo stato.

- Il giocatore si disconnette da una sessione di gioco: Amazon GameLift Servers I messaggi di registro locali dovrebbero indicare che il server di gioco effettua chiamate`RemovePlayerSession()`. Le chiamate AWS CLI a `DescribePlayerSessions()` riflettono una modifica dello stato da `Active` a `Completed`. È inoltre possibile chiamare `DescribeGameSessions()` per verificare che il numero di giocatori della sessione di gioco corrente diminuisce di uno.
- La sessione di gioco termina: Amazon GameLift Servers I messaggi di registro locali dovrebbero mostrare che il server di gioco effettua chiamate`TerminateGameSession()`.

  > **Note**
  >
  > La guida precedente era quella di chiamare al `TerminateGameSession()` termine di una sessione di gioco. Questo metodo è obsoleto con Amazon GameLift Servers Server SDK v4.0.1. Per informazioni, consulta Termina una sessione di gioco.

- Il processo del server è terminato: Amazon GameLift Servers I messaggi di registro locali dovrebbero mostrare che il server di gioco effettua chiamate`ProcessEnding()`. Una AWS CLI chiamata a `DescribeGameSessions()` dovrebbe riflettere una modifica dello stato da `Active` a `Terminated` (o`Terminating`).

Varianti con locale

Quando si utilizza Amazon GameLift Servers Locale, tieni presente quanto segue:

- A differenza del Amazon GameLift Servers servizio web, Local non tiene traccia dello stato di salute del server e avvia il `onProcessTerminate` callback. Local interrompe semplicemente la registrazione dei rapporti sullo stato per il server di gioco.
- Per le chiamate all' AWS SDK, la flotta non IDs viene convalidata e può essere qualsiasi valore di stringa che soddisfi i requisiti del parametro (). `^fleet-\S+`
- Le sessioni di gioco IDs create con Local hanno una struttura diversa. Includono la stringa `local`, come mostrato qui:

```
arn:aws:gamelift:local::gamesession/fleet-123/gsess-56961f8e-
db9c-4173-97e7-270b82f0daa6
```

Aggiungere il FlexMatch matchmaking

Utilizzalo Amazon GameLift Servers FlexMatch per aggiungere funzionalità di matchmaking tra giocatori ai giochi Amazon GameLift Servers ospitati. Puoi utilizzarlo FlexMatch con server di gioco personalizzati o Amazon GameLift ServersRealtime.

FlexMatch associa il servizio di abbinamento a un motore di regole personalizzabili. Progetti come abbinare i giocatori in base agli attributi dei giocatori e alle modalità di gioco più adatte al tuo gioco. FlexMatchgestisce le fasi di valutazione dei giocatori che stanno cercando una partita, organizza le partite con una o più squadre e avvia le sessioni di gioco per ospitare le partite.

Per utilizzare il FlexMatch servizio completo, è necessario che le risorse di hosting siano configurate con code. Amazon GameLift Serversutilizza le code per individuare le migliori sedi di hosting possibili per i giochi in diverse regioni e tipi di computer. In particolare, le Amazon GameLift Servers code possono utilizzare i dati di latenza, se forniti dai client di gioco, per organizzare le sessioni di gioco in modo che i giocatori abbiano la latenza più bassa possibile durante il gioco.

Per ulteriori informazioni su FlexMatch come includere assistenza dettagliata sull'integrazione del matchmaking nei tuoi giochi, consulta questi argomenti della Guida per gli sviluppatori: Amazon GameLift ServersFlexMatch

- Come funziona Amazon GameLift ServersFlexMatch
- FlexMatchfasi di integrazione

Ottieni dati sulla flotta per un Amazon GameLift Servers esempio

In alcune situazioni la build o Amazon GameLift Servers Realtime lo script del gioco personalizzato potrebbero richiedere informazioni sulla flotta. Amazon GameLift Servers Ad esempio, la build o lo script del gioco potrebbero includere codice per:

- Monitora l'attività in base ai dati della flotta.
- Aggiusta le metriche per tracciare l'attività in base ai dati della flotta. (Molti giochi utilizzano questi dati per LiveOps le attività.)
- Fornisci dati pertinenti ai servizi di gioco personalizzati, ad esempio per matchmaking, scalabilità aggiuntiva della capacità o test.

Le informazioni sulla flotta sono disponibili come file JSON su ogni istanza nelle seguenti posizioni:

- Windows: `C:\GameMetadata\gamelift-metadata.json`
- Linux: `/local/gamemetadata/gamelift-metadata.json`

Il `gamelift-metadata.json` file include gli attributi di una risorsa del Amazon GameLift Servers parco veicoli.

File JSON di esempio:

```
{
    "buildArn":"arn:aws:gamelift:us-west-2:123456789012:build/
build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
    "buildId":"build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
    "fleetArn":"arn:aws:gamelift:us-west-2:123456789012:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "fleetDescription":"Test fleet for Really Fun Game v0.8",
    "fleetId":"fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "name":"ReallyFunGameTestFleet08",
    "fleetType":"ON_DEMAND",
    "instanceRoleArn":"arn:aws:iam::123456789012:role/S3AccessForGameLift",
    "instanceType":"c5.large",
    "serverLaunchPath":"/local/game/reallyfungame.exe"
}
```

Integrazione di giochi con Amazon GameLift ServersRealtime

Questo argomento fornisce una panoramica della Amazon GameLift Servers Realtime soluzione Managed Amazon GameLift Servers with. La panoramica spiega quando questa soluzione è adatta al tuo gioco e come Amazon GameLift Servers Realtime supporta i giochi multiplayer.

> **Tip**
>
> Inizia subito con Amazon GameLift Servers

Cosa sono Realtime i server?

Realtimei server sono server di ready-to-go gioco leggeri che ti Amazon GameLift Servers consentono di utilizzarli con i tuoi giochi multiplayer. Realtimei server rimuovono il processo di

sviluppo, test e distribuzione di un server di gioco personalizzato. Questa soluzione può aiutare a ridurre al minimo il tempo e l'impegno necessari per completare il gioco.

Funzionalità principali

- Stack di rete completo per l'interazione tra client e server di gioco
- Funzionalità di base del server di gioco
- Logica del server personalizzabile
- Aggiornamenti in tempo reale Realtime delle configurazioni e della logica del server
- FlexMatchmatchmaking
- Controllo flessibile delle risorse di hosting

Configura Realtime i server creando una flotta e fornendo uno script di configurazione.

Come Amazon GameLift ServersRealtime gestisce le sessioni di gioco

Puoi aggiungere una logica personalizzata per la gestione delle sessioni di gioco inserendola nello Realtime script. Puoi scrivere codice per accedere a oggetti specifici del server, aggiungere logica basata sugli eventi utilizzando callback o aggiungere logica basata su scenari senza eventi.

RealtimeCome interagiscono client e server

Durante una sessione di gioco, i client di gioco interagiscono inviando messaggi al Realtime server tramite un servizio di backend. Il servizio di backend inoltra quindi i messaggi tra i client di gioco per scambiare attività, stato del gioco e dati di gioco pertinenti.

Inoltre, puoi personalizzare l'interazione tra client e server aggiungendo una logica di gioco allo script di Realtime. Con una logica di gioco personalizzata, un Realtime server potrebbe implementare callback per avviare risposte basate sugli eventi.

Protocollo di comunicazione

Realtimei server e i client di gioco connessi comunicano attraverso due canali: una connessione TCP per una consegna affidabile e un canale UDP per una consegna rapida. Durante la creazione di messaggi, i client di gioco scelgono quale protocollo utilizzare in base alla natura del messaggio. La distribuzione dei messaggi è impostata su UDP per impostazione predefinita. Se un canale UDP non è disponibile, Amazon GameLift Servers invia messaggi utilizzando TCP come fallback.

Contenuto del messaggio

Il contenuto del messaggio è composto da due elementi: un codice dell'operazione (opCode) e un payload opzionale. L'OPCode di un messaggio identifica una particolare attività del giocatore o un evento di gioco e il payload fornisce dati aggiuntivi relativi al codice operativo. Entrambi questi elementi sono definiti dagli sviluppatori. Il tuo client di gioco agisce in base agli OpCode presenti nei messaggi che riceve.

Gruppi di giocatori

Amazon GameLift ServersRealtimefornisce funzionalità per gestire gruppi di giocatori. Per impostazione predefinita, Amazon GameLift Servers inserisce tutti i giocatori che si connettono a una partita in un gruppo «tutti i giocatori». Inoltre, gli sviluppatori possono configurare altri gruppi per i propri giochi e i giocatori possono essere membri di più gruppi contemporaneamente. I membri del gruppo possono inviare messaggi e condividere dati di gioco con tutti i giocatori del gruppo. I gruppi possono essere utilizzati ad esempio per configurare team di giocatori e gestire le comunicazioni del team.

Amazon GameLift ServersRealtimecon certificati TLS

Con Amazon GameLift ServersRealtime, l'autenticazione del server e la crittografia dei pacchetti di dati sono integrate nel servizio. Queste funzionalità di sicurezza sono abilitate quando si attiva la generazione di certificati TLS. Quando un client di gioco tenta di connettersi a un Realtime server, il server risponde automaticamente con il certificato TLS, che il client convalida. Amazon GameLift Serversgestisce la crittografia utilizzando TLS per la comunicazione TCP (WebSockets) e DTLS per il traffico UDP.

Personalizzazione di un server Realtime

Un Realtime server funziona come un server di inoltro stateless. Il Realtime server inoltra pacchetti di messaggi e dati di gioco tra i client di gioco collegati al gioco. Tuttavia, il Realtime server non valuta i messaggi, elabora i dati né esegue alcuna logica di gioco. Utilizzato in questo modo, ogni client di gioco mantiene la propria visione dello stato del gioco e fornisce aggiornamenti agli altri giocatori tramite il server di inoltro. Ogni client di gioco è responsabile dell'integrazione di questi aggiornamenti e della riconciliazione delle proprie statistiche di gioco.

Puoi personalizzare i tuoi server aggiungendo funzionalità di Realtime script. Con la logica di gioco, ad esempio, puoi creare un gioco stateful con una visualizzazione autorevole dello stato del gioco da parte del server.

Amazon GameLift Serversdefinisce una serie di callback lato server per gli script. Realtime Implementa queste chiamate per aggiungere funzionalità basate su eventi al server. Ad esempio, puoi:

- Autenticare un giocatore quando un client di gioco cerca di connettersi al server.
- Verifica se un giocatore può unirsi a un gruppo su richiesta.
- Stabilisci quando recapitare i messaggi da un determinato giocatore o a un giocatore bersaglio, oppure esegui un'ulteriore elaborazione in risposta.
- Avvisa tutti i giocatori quando un giocatore lascia un gruppo o si disconnette dal server.
- Visualizza il contenuto degli oggetti della sessione di gioco o degli oggetti dei messaggi e usa i dati.

Distribuzione e aggiornamento Amazon GameLift ServersRealtime

Uno dei principali vantaggi di Amazon GameLift Servers Realtime è la possibilità di aggiornare gli script in qualsiasi momento. Quando aggiorni uno script, Amazon GameLift Servers distribuisce la nuova versione a tutte le risorse di hosting in pochi minuti. Dopo aver Amazon GameLift Servers distribuito il nuovo script, tutte le nuove sessioni di gioco create dopo quel momento utilizzeranno la nuova versione dello script. (Le sessioni di gioco esistenti continueranno a utilizzare la versione originale).

Inizia a integrare il tuo gioco con Amazon GameLift ServersRealtime:

- Integrazione di un client di gioco per Amazon GameLift ServersRealtime
- Creazione di uno Realtime script

Integrazione di un client di gioco per Amazon GameLift ServersRealtime

Questo argomento descrive come preparare il client di gioco per essere in grado di partecipare a sessioni di gioco ospitate da Amazon GameLift Servers.

Sono disponibili due set di attività necessarie per preparare il client di gioco:

- Configurare il client di gioco per acquisire informazioni su giochi esistenti, richiedere l'abbinamento, avviare nuove sessioni di gioco e prenotare slot delle sessioni di gioco per un giocatore.
- Abilitare il client di gioco a unirsi a una sessione di gioco in hosting su un server Realtime e scambiare messaggi.

Trova o crea sessioni di gioco e sessioni per giocatori

Configurare il client di gioco per trovare o avviare le sessioni di gioco, richiedere l'abbinamento FlexMatch prenotare spazi per giocatori in un gioco mediante la creazione di sessioni giocatore. Come procedura consigliata, crea un servizio di backend e utilizzalo per effettuare richieste dirette al Amazon GameLift Servers servizio quando attivato da un'azione del client di gioco. Il servizio di backend inoltra quindi le risposte pertinenti al client di gioco.

1. Aggiungi l' AWS SDK al tuo client di gioco, inizializza un Amazon GameLift Servers client e configuralo per utilizzare le risorse di hosting presenti nelle tue flotte e nelle tue code. L' AWS SDK è disponibile in diverse lingue; consulta la. Amazon GameLift Servers SDKs Per i servizi client di gioco
2. Aggiungi Amazon GameLift Servers funzionalità al tuo servizio di backend. Per istruzioni più dettagliate, vedere Aggiungi Amazon GameLift Servers al tuo client di gioco e Aggiungere FlexMatch matchmaking. La best practice è utilizzare posizionamenti delle sessioni di gioco per creare nuove sessioni di gioco. Questo metodo ti consente di sfruttare appieno la Amazon GameLift Servers capacità di effettuare nuove sessioni di gioco in modo rapido e intelligente, oltre a utilizzare i dati sulla latenza dei giocatori per ridurre al minimo il ritardo di gioco. Come minimo, il servizio di backend deve essere in grado di richiedere nuove sessioni di gioco e gestire i dati delle sessioni di gioco in risposta. È anche possibile aggiungere funzionalità per cercare e ottenere informazioni su sessioni di gioco esistenti e richiedere sessioni dei giocatori che effettivamente prenotano uno slot del giocatore in una sessione di gioco esistente.
3. Trasmettere le informazioni di connessione al client di gioco. Il servizio di backend riceve gli oggetti della sessione di gioco e della sessione dei giocatori in risposta alle richieste al Amazon GameLift Servers servizio. Questi oggetti contengono informazioni, in particolare i dettagli di connessione (indirizzo IP e porta) e l'ID della sessione del giocatore, di cui il client di gioco ha bisogno per connettersi alla sessione di gioco in esecuzione su un server Realtime.

Connect ai giochi su Amazon GameLift ServersRealtime

Abilitare il client di gioco per connettersi direttamente a una sessione di gioco in hosting su un server Realtime, e scambiare messaggi con il server e con altri giocatori.

1. Scarica l'SDK del client Amazon GameLift ServersRealtime, crealo e aggiungilo al tuo progetto di client di gioco. Vedi il file README per ulteriori informazioni sui requisiti dell'SDK e sulle istruzioni su come creare librerie client.

2. Chiama Client() con una configurazione client che specifica il tipo di client/server connessione da utilizzare.

> **Note**
>
> Per la connessione a un server Realtime in esecuzione su un parco istanze protetto con un certificato TLS, devi specificare un tipo di connessione protetta.

3. Aggiungere le seguenti funzionalità al client di gioco. Per ulteriori informazioni, consulta Amazon GameLift ServersRealtimeriferimento all'API client (C#).
   - Connettersi e disconnettersi da un gioco
     - Connect()
     - Disconnect()
   - Inviare messaggi a destinatari target
     - SendMessage()
   - Ricevere ed elaborare messaggi
     - OnDataReceived()
   - Entrare a far parte di gruppi e lasciare gruppi di giocatori
     - JoinGroup()
     - RequestGroupMembership()
     - LeaveGroup()
4. Configurare gestori di eventi per le chiamate del client in base alle esigenze. Consultare Amazon GameLift ServersRealtimeriferimento all'API client (C#): callback asincroni.

Quando si utilizzano parchi istanze Realtime con la generazione di certificati TLS abilitata, il server viene automaticamente autenticato utilizzando il certificato TLS. Il traffico TCP e UDP viene crittografato in transito per fornire sicurezza a livello di trasporto. Il traffico TCP viene crittografato con TLS 1.2 e il traffico UDP viene crittografato con DTLS 1.2.

Esempi di client di gioco

Client di base in tempo reale (C#)

Questo esempio illustra un'integrazione di base del client di gioco con il client SDK (C#) for. Amazon GameLift Servers Realtime Come illustrato, l'esempio inizializza un oggetto client Realtime, imposta

gestori di eventi e implementa il callback lato client, si connette a un server Realtime, invia un messaggio e si disconnette.

```
using System;
using System.Text;
using Aws.GameLift.Realtime;
using Aws.GameLift.Realtime.Event;
using Aws.GameLift.Realtime.Types;

namespace Example
{
    /**
     * An example client that wraps the client SDK for Amazon GameLift Servers Realtime
     *
     * You can redirect logging from the SDK by setting up the LogHandler as such:
     * ClientLogger.LogHandler = (x) => Console.WriteLine(x);
     *
     */
    class RealTimeClient
    {
        public Aws.GameLift.Realtime.Client Client { get; private set; }

        // An opcode defined by client and your server script that represents a custom
 message type
        private const int MY_TEST_OP_CODE = 10;

        /// Initialize a client for Amazon GameLift Servers Realtime and connect to a
 player session.
        /// <param name="endpoint">The DNS name that is assigned to Realtime server</
param>
        /// <param name="remoteTcpPort">A TCP port for the Realtime server</param>
        /// <param name="listeningUdpPort">A local port for listening to UDP traffic</
param>
        /// <param name="connectionType">Type of connection to establish between client
 and the Realtime server</param>
        /// <param name="playerSessionId">The player session ID that is assigned to the
 game client for a game session </param>
        /// <param name="connectionPayload">Developer-defined data to be used during
 client connection, such as for player authentication</param>
       public RealTimeClient(string endpoint, int remoteTcpPort, int listeningUdpPort,
 ConnectionType connectionType,
                      string playerSessionId, byte[] connectionPayload)
        {
```

```
            // Create a client configuration to specify a secure or unsecure connection
type
            // Best practice is to set up a secure connection using the connection type
RT_OVER_WSS_DTLS_TLS12.
            ClientConfiguration clientConfiguration = new ClientConfiguration()
     {
                // C# notation to set the field ConnectionType in the new instance of
ClientConfiguration
         ConnectionType = connectionType
     };

            // Create a Realtime client with the client configuration
            Client = new Client(clientConfiguration);

            // Initialize event handlers for the Realtime client
            Client.ConnectionOpen += OnOpenEvent;
            Client.ConnectionClose += OnCloseEvent;
            Client.GroupMembershipUpdated += OnGroupMembershipUpdate;
            Client.DataReceived += OnDataReceived;

            // Create a connection token to authenticate the client with the Realtime
server
            // Player session IDs can be retrieved using AWS SDK for Amazon GameLift
Servers
            ConnectionToken connectionToken = new ConnectionToken(playerSessionId,
connectionPayload);

            // Initiate a connection with the Realtime server with the given connection
information
            Client.Connect(endpoint, remoteTcpPort, listeningUdpPort, connectionToken);
       }

       public void Disconnect()
       {
           if (Client.Connected)
           {
               Client.Disconnect();
           }
       }

       public bool IsConnected()
       {
           return Client.Connected;
       }
```

```
        /// <summary>
        /// Example of sending to a custom message to the server.
        ///
        /// Server could be replaced by known peer Id etc.
        /// </summary>
        /// <param name="intent">Choice of delivery intent i.e. Reliable, Fast etc. </
param>
        /// <param name="payload">Custom payload to send with message</param>
        public void SendMessage(DeliveryIntent intent, string payload)
        {
            Client.SendMessage(Client.NewMessage(MY_TEST_OP_CODE)
                .WithDeliveryIntent(intent)
                .WithTargetPlayer(Constants.PLAYER_ID_SERVER)
                .WithPayload(StringToBytes(payload)));
        }

        /**
         * Handle connection open events
         */
        public void OnOpenEvent(object sender, EventArgs e)
        {
        }

        /**
         * Handle connection close events
         */
        public void OnCloseEvent(object sender, EventArgs e)
        {
        }

        /**
         * Handle Group membership update events
         */
        public void OnGroupMembershipUpdate(object sender, GroupMembershipEventArgs e)
        {
        }

        /**
         *  Handle data received from the Realtime server
         */
        public virtual void OnDataReceived(object sender, DataReceivedEventArgs e)
        {
            switch (e.OpCode)
```

```
            {
                // handle message based on OpCode
                default:
                    break;
            }
        }

        /**
         * Helper method to simplify task of sending/receiving payloads.
         */
        public static byte[] StringToBytes(string str)
        {
            return Encoding.UTF8.GetBytes(str);
        }

        /**
         * Helper method to simplify task of sending/receiving payloads.
         */
        public static string BytesToString(byte[] bytes)
        {
            return Encoding.UTF8.GetString(bytes);
        }
    }
}
```

Creazione di uno Realtime script

Da utilizzare Amazon GameLift Servers Realtime per il gioco, è necessario fornire uno script (sotto forma di JavaScript codice) per configurare e, facoltativamente, personalizzare una flotta di Amazon GameLift ServersRealtime. In questo argomento vengono illustrati i passaggi fondamentali per la creazione di uno script Realtime. Una volta che lo script è pronto, caricalo nel servizioAmazon GameLift Servers e usalo per creare un parco istanze (consulta Implementa uno script per Amazon GameLift ServersRealtime).

Per preparare uno script da utilizzare con Amazon GameLift ServersRealtime, aggiungi la seguente funzionalità allo Realtime script.

Gestisci il ciclo di vita della sessione di gioco (richiesto)

Come minimo, uno script Realtime deve includere la funzione `Init()`, che prepara il server Realtime per avviare una sessione di gioco. Inoltre, è consigliabile fornire un modo per terminare le sessioni

di gioco, così da assicurare che le nuove sessioni di gioco possano continuare a essere avviate sul parco istanze.

La funzione di chiamata `Init()`, quando usata, viene passata a un oggetto di sessione di Realtime, che contiene un'interfaccia per il server Realtime. Per ulteriori informazioni su questa interfaccia, consulta Amazon GameLift ServersRealtimeinterfaccia.

Per terminare in modo corretto una sessione di gioco, lo script deve anche chiamare la funzione `session.processEnding` del server Realtime. Ciò richiede alcuni meccanismi per determinare quando terminare una sessione. Il codice script di esempio illustra un semplice meccanismo che controlla le connessioni dei giocatori e attiva la terminazione della sessione di gioco quando nessun giocatore è connesso alla sessione per un determinato intervallo di tempo.

Amazon GameLift ServersRealtimecon la configurazione di base (inizializzazione e terminazione dei processi del server) funzionano essenzialmente come server di inoltro stateless. Il server Realtime inoltra messaggi e dati di gioco tra i client di gioco collegati al gioco, ma non esegue operazioni indipendenti per elaborare i dati o per eseguire una logica. È anche possibile aggiungere una logica di gioco, attivata da eventi di gioco o da altri meccanismi, in base alle esigenze del gioco medesimo.

Aggiungi la logica di gioco lato server (opzionale)

È anche possibile aggiungere logica di gioco allo script Realtime. Ad esempio, è possibile eseguire una o tutte le seguenti operazioni. Il codice di esempio di script fornisce illustrazione. Consultare Riferimento allo script per Amazon GameLift ServersRealtime.

- Aggiunta di logica basata sugli eventi. Implementa le funzioni di chiamata per rispondere a eventi client-server. Per un elenco completo di chiamate, consulta Richiamate di script per Amazon GameLift ServersRealtime.
- Attivazione della logica mediante l'invio di messaggi al server. Crea un set di codici di operazione speciali per i messaggi inviati dai client di gioco al server e aggiunge funzioni per gestire la ricezione Utilizza la chiamata on`Message` e analizza il contenuto del messaggio tramite l'interfaccia `gameMessage` (consulta gameMessage.opcode).
- Abilita la logica di gioco per accedere alle altre AWS risorse. Per informazioni dettagliate, consultare Comunica con altre AWS risorse delle tue flotte.
- Consenti alla logica di gioco di accedere alle informazioni sulla flotta per l'istanza su cui è in esecuzione. Per informazioni dettagliate, consultare Ottieni dati sulla flotta per un Amazon GameLift Servers esempio.

Amazon GameLift ServersRealtimeesempio di script

Questo esempio illustra uno script di base necessario per la distribuzione e una Amazon GameLift Servers Realtime logica personalizzata. Contiene la funzione `Init()` richiesta e utilizza un meccanismo timer per attivare la terminazione delle sessioni di gioco in base all'intervallo di tempo in cui nessun giocatore si è connesso. Include anche alcuni hook per logica personalizzata, tra cui alcune implementazioni di chiamate.

```
// Example Realtime Server Script
'use strict';

// Example override configuration
const configuration = {
    pingIntervalTime: 30000,
    maxPlayers: 32
};

// Timing mechanism used to trigger end of game session. Defines how long, in
 milliseconds, between each tick in the example tick loop
const tickTime = 1000;

// Defines how to long to wait in Seconds before beginning early termination check in
 the example tick loop
const minimumElapsedTime = 120;

var session;                        // The Realtime server session object
var logger;                         // Log at appropriate level
 via .info(), .warn(), .error(), .debug()
var startTime;                      // Records the time the process started
var activePlayers = 0;              // Records the number of connected players
var onProcessStartedCalled = false; // Record if onProcessStarted has been called

// Example custom op codes for user-defined messages
// Any positive op code number can be defined here. These should match your client
 code.
const OP_CODE_CUSTOM_OP1 = 111;
const OP_CODE_CUSTOM_OP1_REPLY = 112;
const OP_CODE_PLAYER_ACCEPTED = 113;
const OP_CODE_DISCONNECT_NOTIFICATION = 114;

// Example groups for user-defined groups
// Any positive group number can be defined here. These should match your client code.
// When referring to user-defined groups, "-1" represents all groups, "0" is reserved.
```

```
const RED_TEAM_GROUP = 1;
const BLUE_TEAM_GROUP = 2;

// Called when game server is initialized, passed server's object of current session
function init(rtSession) {
    session = rtSession;
    logger = session.getLogger();
}

// On Process Started is called when the process has begun and we need to perform any
// bootstrapping.  This is where the developer should insert any code to prepare
// the process to be able to host a game session, for example load some settings or set
 state
//
// Return true if the process has been appropriately prepared and it is okay to invoke
 the
// GameLift ProcessReady() call.
function onProcessStarted(args) {
    onProcessStartedCalled = true;
    logger.info("Starting process with args: " + args);
    logger.info("Ready to host games...");

    return true;
}

// Called when a new game session is started on the process
function onStartGameSession(gameSession) {
    // Complete any game session set-up

    // Set up an example tick loop to perform server initiated actions
    startTime = getTimeInS();
    tickLoop();
}

// Handle process termination if the process is being terminated by Amazon GameLift
 Servers
// You do not need to call ProcessEnding here
function onProcessTerminate() {
    // Perform any clean up
}

// Return true if the process is healthy
function onHealthCheck() {
    return true;
```

```
}

// On Player Connect is called when a player has passed initial validation
// Return true if player should connect, false to reject
function onPlayerConnect(connectMsg) {
    // Perform any validation needed for connectMsg.payload, connectMsg.peerId
    return true;
}

// Called when a Player is accepted into the game
function onPlayerAccepted(player) {
    // This player was accepted -- let's send them a message
    const msg = session.newTextGameMessage(OP_CODE_PLAYER_ACCEPTED, player.peerId,
                                           "Peer " + player.peerId + " accepted");
    session.sendReliableMessage(msg, player.peerId);
    activePlayers++;
}

// On Player Disconnect is called when a player has left or been forcibly terminated
// Is only called for players that actually connected to the server and not those
 rejected by validation
// This is called before the player is removed from the player list
function onPlayerDisconnect(peerId) {
    // send a message to each remaining player letting them know about the disconnect
    const outMessage = session.newTextGameMessage(OP_CODE_DISCONNECT_NOTIFICATION,
                                              session.getServerId(),
                                              "Peer " + peerId + " disconnected");
    session.getPlayers().forEach((player, playerId) => {
        if (playerId != peerId) {
            session.sendReliableMessage(outMessage, playerId);
        }
    });
    activePlayers--;
}

// Handle a message to the server
function onMessage(gameMessage) {
    switch (gameMessage.opCode) {
      case OP_CODE_CUSTOM_OP1: {
        // do operation 1 with gameMessage.payload for example sendToGroup
        const outMessage = session.newTextGameMessage(OP_CODE_CUSTOM_OP1_REPLY,
 session.getServerId(), gameMessage.payload);
        session.sendGroupMessage(outMessage, RED_TEAM_GROUP);
        break;
```

```
      }
    }
}

// Return true if the send should be allowed
function onSendToPlayer(gameMessage) {
    // This example rejects any payloads containing "Reject"
    return (!gameMessage.getPayloadAsText().includes("Reject"));
}

// Return true if the send to group should be allowed
// Use gameMessage.getPayloadAsText() to get the message contents
function onSendToGroup(gameMessage) {
    return true;
}

// Return true if the player is allowed to join the group
function onPlayerJoinGroup(groupId, peerId) {
    return true;
}

// Return true if the player is allowed to leave the group
function onPlayerLeaveGroup(groupId, peerId) {
    return true;
}

// A simple tick loop example
// Checks to see if a minimum amount of time has passed before seeing if the game has
 ended
async function tickLoop() {
    const elapsedTime = getTimeInS() - startTime;
    logger.info("Tick... " + elapsedTime + " activePlayers: " + activePlayers);

    // In Tick loop - see if all players have left early after a minimum period of time
 has passed
    // Call processEnding() to terminate the process and quit
    if ( (activePlayers == 0) && (elapsedTime > minimumElapsedTime)) {
        logger.info("All players disconnected. Ending game");
        const outcome = await session.processEnding();
        logger.info("Completed process ending with: " + outcome);
        process.exit(0);
    }
    else {
        setTimeout(tickLoop, tickTime);
```

```
    }
}

// Calculates the current time in seconds
function getTimeInS() {
    return Math.round(new Date().getTime()/1000);
}

exports.ssExports = {
    configuration: configuration,
    init: init,
    onProcessStarted: onProcessStarted,
    onMessage: onMessage,
    onPlayerConnect: onPlayerConnect,
    onPlayerAccepted: onPlayerAccepted,
    onPlayerDisconnect: onPlayerDisconnect,
    onSendToPlayer: onSendToPlayer,
    onSendToGroup: onSendToGroup,
    onPlayerJoinGroup: onPlayerJoinGroup,
    onPlayerLeaveGroup: onPlayerLeaveGroup,
    onStartGameSession: onStartGameSession,
    onProcessTerminate: onProcessTerminate,
    onHealthCheck: onHealthCheck
};
```

Implementazione del software del server di gioco per l'hosting Amazon GameLift Servers

Implementa il software del tuo server di gioco multiplayer installandolo sulle tue risorse di hosting, avviando i processi del server di gioco e preparandoli a ospitare partite per i giocatori. I passaggi per preparare il software del server di gioco per l'implementazione dipendono dal tipo di Amazon GameLift Servers soluzione che stai utilizzando. In tutti gli scenari, il software del server di gioco distribuito interagisce con il Amazon GameLift Servers servizio per gestire il posizionamento delle sessioni di gioco e comunicare i dettagli di connessione per un client di gioco.

Il software del server di gioco deve prima essere integrato conAmazon GameLift Servers, come descritto in. Preparazione dei giochi per Amazon GameLift Servers

Gli argomenti di questa sezione forniscono indicazioni su come preparare il software per la distribuzione nei seguenti scenari.

- Software per server di gioco personalizzato
  - Per l' EC2 hosting Amazon GameLift Servers gestito
  - Per l'hosting di container Amazon GameLift Servers gestiti
  - Per l'hosting Amazon GameLift Servers ovunque
- Amazon GameLift ServersRealtimescript personalizzato per l' EC2 hosting Amazon GameLift Servers gestito

Argomenti

Implementa una build di server personalizzata per Amazon GameLift Servers hosting

Dopo aver integrato il server di gioco con Amazon GameLift Servers (vediPreparazione dei giochi per Amazon GameLift Servers), installa il software del server di gioco sulle tue risorse di calcolo per

l'hosting. Questo processo varia a seconda del tipo di Amazon GameLift Servers hosting che stai utilizzando.

Implementa per l'hosting gestito

Se stai usando Amazon GameLift Servers EC2 hosting gestito, devi impacchettare il software del server di gioco e caricarlo su Amazon GameLift Servers. Quando crei una flotta gestita, Amazon GameLift Servers la distribuisce automaticamente in ogni istanza del parco macchine.

Gli argomenti di questa sezione descrivono come impacchettare i file di build per il caricamento, creare uno script di installazione della build opzionale e quindi caricare i file utilizzando AWS Command Line Interface (AWS CLI) o AWS SDK.

Implementa l'hosting for Anywhere

Se stai utilizzando Amazon GameLift Servers Anywhere dispone di flotte di servizi di hosting autogestiti, è tua responsabilità installare il software del server di gioco su ogni computer del parco computer e mantenerlo aggiornato.

Quando un processo integrato del server di gioco inizia a funzionare, si inizializza automaticamente e stabilisce la comunicazione con Amazon GameLift Servers servizio. Il processo del server avvia le sessioni di gioco in base alle istruzioni di Amazon GameLift Servers e riporta l'attività al servizio.

Argomenti

- Creare un pacchetto dei file della build di gioco
- Aggiungere uno script di installazione della build
- Crea un Amazon GameLift Servers crea risorse per l'hosting gestito
- Aggiorna una build del server di gioco per Amazon GameLift Servers hosting gestito

Creare un pacchetto dei file della build di gioco

Prima di caricare il server di gioco configurato su Amazon GameLift Servers, impacchetta i file di build del gioco in una directory di compilazione. Questo processo è un requisito quando si ospita con una flotta EC2 gestita ed è una buona pratica quando si ospita con una flotta Anywhere. La directory di build dovrebbe includere tutti i componenti necessari per far funzionare i server di gioco e ospitare sessioni di gioco. Ciò potrebbe includere quanto segue:

- File binari del server di gioco: i file binari necessari per eseguire il server di gioco. Una build può includere file binari per più server di gioco creati per funzionare sulla stessa piattaforma. Per un elenco delle piattaforme supportate, consultaOttieni strumenti Amazon GameLift Servers di sviluppo.
- Dipendenze: tutti i file dipendenti necessari per l'esecuzione degli eseguibili del server di gioco. Esempi di file dipendenti includono gli asset, i file di configurazione e le librerie dipendenti.

> **Note**
>
> Per le build di gioco create con l'SDK del server per Amazon GameLift Servers per C++ (compresi quelli creati con il plugin Unreal), includi la DLL OpenSSL per la stessa versione di OpenSSL con cui hai creato l'SDK del server. Per maggiori dettagli, consulta il file README dell'SDK del server.

- Script di installazione (opzionale): un file di script per gestire le attività relative all'installazione della build del gioco Amazon GameLift Servers server di hosting. Posiziona questo file nella radice della directory di compilazione. Amazon GameLift Servers esegue lo script di installazione come parte della creazione della flotta.

Puoi configurare qualsiasi applicazione della tua build, incluso lo script di installazione, per accedere alle tue risorse in modo sicuro su altri AWS servizi. Per informazioni su come eseguire questa operazione, consultaComunica con altre AWS risorse delle tue flotte.

Dopo aver impacchettato i file di build, assicurati che il server di gioco possa funzionare su un'installazione pulita del sistema operativo di destinazione per verificare che tutte le dipendenze richieste siano incluse e che lo script di installazione sia accurato.

Aggiungere uno script di installazione della build

Crea uno script di installazione per il sistema operativo (OS) della build del tuo gioco:

- Windows: crea un file batch denominato`install.bat`.
- Linux: crea un file di script di shell denominato`install.sh`.

Quando crei uno script di installazione, considera quanto segue:

- Lo script non può accettare alcun input da parte dell'utente.

- Amazon GameLift Servers installa la build e ricrea le directory dei file nel pacchetto di build su un server di hosting nelle seguenti posizioni:
  - Flotte Windows: `C:\game`
  - Flotte Linux: `/local/game`
- Durante il processo di installazione per le flotte Linux, l'utente run-as ha accesso limitato alla struttura dei file dell'istanza. Questo utente dispone dei diritti completi sulla directory in cui sono installati i file di build. Se lo script di installazione esegue azioni che richiedono le autorizzazioni di amministratore, specifica l'accesso da amministratore utilizzandosudo. Per impostazione predefinita, l'utente run-as for Windows Fleets dispone delle autorizzazioni di amministratore. Gli errori di autorizzazione correlati allo script di installazione generano un messaggio di evento che indica un problema dello script.
- Su Linux, Amazon GameLift Servers supporta i comuni linguaggi di interpretazione della shell come bash. Aggiungi uno shebang (ad esempio `#!/bin/bash`) all'inizio dello script di installazione. Per verificare il supporto per i comandi di shell preferiti, accedi in remoto a un'istanza Linux attiva e apri un prompt della shell. Per ulteriori informazioni, consulta Connessione remota a Amazon GameLift Servers flotte di istanze.
- Lo script di installazione non può fare affidamento su una connessione peering VPC. Una connessione peering VPC è disponibile solo dopo Amazon GameLift Servers installa la build su una flotta di istanze.

Example Windows installa il file bash

Questo `install.bat` file di esempio installa i componenti di runtime di Visual C++ necessari per il server di gioco e scrive i risultati in un file di registro. Lo script include il file del componente nel pacchetto di build alla radice.

```
vcredist_x64.exe /install /quiet /norestart /log c:\game\vcredist_2013_x64.log
```

Example Script della shell di installazione di Linux

Questo `install.sh` file di esempio utilizza bash nello script di installazione e scrive i risultati in un file di registro.

```
#!/bin/bash
echo 'Hello World' > install.log
```

Questo `install.sh` file di esempio mostra come utilizzare l' CloudWatch agente Amazon per raccogliere metriche a livello di sistema e personalizzate e gestire la rotazione dei log. Perché Amazon GameLift Servers viene eseguito in un servizio VPC, è necessario concedere Amazon GameLift Servers autorizzazioni per assumere un ruolo AWS Identity and Access Management (IAM) per tuo conto. Per consentire Amazon GameLift Servers per assumere un ruolo, crea un ruolo che includa la politica AWS `CloudWatchAgentAdminPolicy` gestita e usa quel ruolo quando crei una flotta.

```
sudo yum install -y amazon-cloudwatch-agent
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-
latest-7.noarch.rpm
sudo yum install -y collectd
cat <<'EOF' > /tmp/config.json
{
    "agent": {
        "metrics_collection_interval": 60,
        "run_as_user": "root",
        "credentials": {
            "role_arn": "arn:aws:iam::account#:role/rolename"
        }
    },
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/tmp/log",
                        "log_group_name": "gllog",
                        "log_stream_name": "{instance_id}"
                    }
                ]
            }
        }
    },
    "metrics": {
       "namespace": "GL_Metric",
        "append_dimensions": {
            "ImageId": "${aws:ImageId}",
            "InstanceId": "${aws:InstanceId}",
            "InstanceType": "${aws:InstanceType}"
        },
        "metrics_collected": {
```

```
            // Configure metrics you want to collect.
            // For more information, see Manually create or edit the CloudWatch agent
 configuration file.
        }
    }
}
EOF
sudo mv /tmp/config.json /opt/aws/amazon-cloudwatch-agent/bin/config.json
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
sudo systemctl enable amazon-cloudwatch-agent.service
```

Crea un Amazon GameLift Servers crea risorse per l'hosting gestito

Per creare una build e caricare i file, sono disponibili un paio di opzioni:

- Crea una build da una directory di file. Questa è l'opzione più semplice e più comunemente usata.
- Crea una build con file in Amazon Simple Storage Service (Amazon S3). Con questa opzione, puoi gestire le tue versioni di build in Amazon S3.

Con entrambi i metodi, Amazon GameLift Servers crea una nuova risorsa di compilazione con un ID di build univoco e altri metadati. La build inizia con lo stato Inizializzato. Dopo Amazon GameLift Servers acquisisce i file del server di gioco, la build passa allo stato Pronto.

Quando la build è pronta, puoi distribuirla in una nuova Amazon GameLift Servers flotta. Per ulteriori informazioni, consulta Crea una EC2 flotta Amazon GameLift Servers gestita .When Amazon GameLift Servers configura la nuova flotta, scarica i file di build in ogni istanza della flotta e installa i file di build.

Crea una build da una directory di file

Per creare una build di gioco memorizzata in qualsiasi posizione, inclusa una directory locale, usa il `upload-build` AWS CLI comando. Questo comando crea un nuovo record di build in Amazon GameLift Servers e carica i file da una posizione specificata dall'utente.

Inviare una richiesta di caricamento. In una finestra della riga di comando, immettete il upload-build comando e i parametri seguenti.

```
aws gamelift upload-build \
    --name user-defined name of build \
    --operating-system supported OS \
```

```
    --server-sdk-version server SDK for Amazon GameLift Servers version \
    --build-root build path \
    --build-version user-defined build number \
    --region region name
```

- operating-system— L'ambiente di runtime della build del server di gioco. È necessario specificare un valore del sistema operativo. Non puoi aggiornarlo in un secondo momento.
- server-sdk-version— La versione del Amazon GameLift Servers server SDK con cui è integrato il tuo server di gioco. Se non fornisci un valore, Amazon GameLift Servers utilizza il valore predefinito`4.0.2`. Se specifichi una versione SDK del server non corretta, la build del server di gioco potrebbe non riuscire durante la chiamata `InitSdk` per stabilire una connessione al Amazon GameLift Servers servizio.
- build-root— Il percorso della directory dei file di build.
- name— Un nome descrittivo per la nuova build.
- build-version— I dettagli della versione per i file di build.
- region— La AWS regione in cui vuoi creare la tua build. Crea la build nella regione in cui intendi schierare le flotte. Se stai distribuendo il gioco in più regioni, crea una build in ogni regione.

> **Note**
>
> Visualizza la tua regione predefinita attuale utilizzando il aws configure get region. Per modificare la tua regione predefinita, usa il aws configure set region ***region name***comando.

Examples (Esempi)

```
aws gamelift upload-build \
    --operating-system AMAZON_LINUX_2023 \

    --server-sdk-version "5.0.0" \
    --build-root "~/mygame" \
    --name "My Game Nightly Build" \
    --build-version "build 255" \
    --region us-west-2
```

```
aws gamelift upload-build \
    --operating-system WINDOWS_2016 \
```

```
    --server-sdk-version "5.0.0" \
    --build-root "C:\mygame" \
    --name "My Game Nightly Build" \
    --build-version "build 255" \
    --region us-west-2
```

In risposta alla tua richiesta di caricamento, Amazon GameLift Servers fornisce l'avanzamento del caricamento. In caso di caricamento riuscito, Amazon GameLift Servers restituisce il nuovo ID del record di build. Il tempo di caricamento dipende dalle dimensioni dei file di gioco e dalla velocità di connessione.

Crea una build con file in Amazon S3

Puoi archiviare i tuoi file di build in Amazon S3 e caricarli su Amazon GameLift Servers da lì. Quando crei la build, specifichi la posizione del bucket S3 e Amazon GameLift Servers recupera i file di build direttamente da Amazon S3.

Per creare una risorsa di compilazione

1. Archivia i tuoi file di build in Amazon S3. Crea un file.zip contenente i file di build impacchettati e caricalo in un bucket S3 del tuo. Account AWS Prendi nota dell'etichetta del bucket e del nome del file, ti serviranno per creare un Amazon GameLift Servers costruire.
2. Dare Amazon GameLift Servers accesso ai tuoi file di build. Crea un ruolo IAM seguendo le istruzioni riportate inAccedi a un file di build del gioco in Amazon S3. Dopo aver creato il ruolo, prendi nota dell'Amazon Resource Name (ARN) del nuovo ruolo, ti servirà per creare una build.
3. Crea una build. Usa il Amazon GameLift Servers console o AWS CLI per creare un nuovo record di build. È necessario disporre dell'`PassRole`autorizzazione, come descritto inEsempi di autorizzazioni IAM per Amazon GameLift Servers.

Console

1. Nella Amazon GameLift Servers console, nel pannello di navigazione, scegli Hosting, Builds.
2. Nella pagina Builds, scegli Crea build.
3. Nella pagina Crea build, in Impostazioni build, procedi come segue:
    a. Per Nome, inserite il nome dello script.
    b. In Versione, inserisci una versione. Poiché puoi aggiornare il contenuto di una build, i dati sulla versione possono aiutarti a tenere traccia degli aggiornamenti.

c. Per Sistema operativo (OS), scegli il sistema operativo della build del tuo server di gioco. Non puoi aggiornare questo valore in un secondo momento.

d. Per la build del server di gioco, inserisci l'URI S3 dell'oggetto di build che hai caricato su Amazon S3 e scegli la versione dell'oggetto. Se non ricordi l'URI e la versione dell'oggetto di Amazon S3, scegli Browse S3 e cerca l'oggetto di compilazione.

e. Per il ruolo IAM, scegli il ruolo che hai creato che offre Amazon GameLift Servers accesso al bucket S3 e all'oggetto di compilazione.

4. (Facoltativo) In Tag, aggiungi i tag alla build inserendo le coppie Chiave e Valore.

5. Scegli Create (Crea) .

Amazon GameLift Servers assegna un ID alla nuova build e carica il file.zip designato. Puoi visualizzare la nuova build, incluso lo stato, nella pagina Builds.

AWS CLI

Per definire la nuova build e caricare i file di build del server, usa il `create-build`comando.

1. Apri una finestra della riga di comando e passa a una directory in cui puoi usare AWS CLI.

2. Immettete il seguente create-build comando:

```
aws gamelift create-build \
    --name user-defined name of build \
    --server-sdk-version server SDK for Amazon GameLift Servers version \
    --operating-system supported OS \
    --build-version user-defined build number \
    --storage-location "Bucket"=S3 bucket label,"Key"=Build .zip file
 name,"RoleArn"=Access role ARN} \
    --region region name
```

- name— Un nome descrittivo per la nuova build.
- server-sdk-version— La versione del server SDK per Amazon GameLift Servers che hai usato per integrare il tuo server di gioco con Amazon GameLift Servers. Se non fornisci un valore, Amazon GameLift Servers utilizza il valore predefinito`4.0.2`.
- operating-system— L'ambiente di runtime della build del server di gioco. È necessario specificare un valore del sistema operativo. Non puoi aggiornarlo in un secondo momento.

- build-version— I dettagli della versione per i file di build. Queste informazioni possono essere utili perché ogni nuova versione del server di gioco richiede una nuova risorsa di build.
- storage-location
  - Bucket— Il nome del bucket S3 che contiene la build. Ad esempio, «my_build_files».
  - Key— Il nome del file.zip che contiene i file di build. Ad esempio, «my_game_build_7.0.1, 7.0.2".
  - RoleARN— L'ARN assegnato al ruolo IAM che hai creato. Ad esempio, «arn:aws:iam: :111122223333:role/». GameLiftAccess Per un esempio di policy, consulta Accedi a un file di build del gioco in Amazon S3.
- region— Crea la build nella regione in cui intendi schierare le flotte. AWS Se stai distribuendo il gioco in più regioni, crea una build in ogni regione.

> **Note**
>
> Ti consigliamo di controllare la tua regione predefinita attuale usando il `configure get`comando . Per cambiare la tua regione predefinita, usa il `configure set`comando.

Esempio

```
aws gamelift create-build \
    --operating-system WINDOWS_2016 \
    --storage-location
 "Bucket"="my_game_build_files","Key"="mygame_build_101.zip","RoleArn"="arn:aws:iam::111
gamelift" \
    --name "My Game Nightly Build" \
    --build-version "build 101" \
    --region us-west-2
```

3. Per visualizzare la nuova build, usa il `describe-build`comando.

Aggiorna una build del server di gioco per Amazon GameLift Servers hosting gestito

Quando distribuisci il tuo server di gioco, crea per Amazon GameLift Servers EC2 hosting gestito, carichi il software del server di gioco e crei un Amazon GameLift Servers crea risorse. Dopo aver creato un Amazon GameLift Servers build, puoi aggiornare i metadati della build, ma non puoi aggiornare i file di build stessi. Per distribuire gli aggiornamenti sul tuo server di gioco, carica i file aggiornati e creane uno nuovo Amazon GameLift Servers costruisci usando il AWS CLI comando di `upload-build`comando. In alternativa, puoi usare il `create-build`comando per caricare una nuova build da un bucket Amazon S3 che controlli. Quindi distribuisci la nuova build creando una nuova flotta per essa.

Puoi aggiornare i metadati di una build, inclusi il nome e la descrizione. Per queste attività, usa il Amazon GameLift Servers console o il `update-build` AWS CLI comando.

Automatizza gli aggiornamenti della build del gioco

Segui questi suggerimenti per automatizzare e semplificare il processo di aggiornamento delle build dei server di gioco per Amazon GameLift Servers flotte gestite:

- Usa le code delle sessioni di gioco e sostituisci le flotte secondo necessità. Quando si inviano richieste di sessioni di gioco a Amazon GameLift Servers, specifica una coda per le sessioni di gioco anziché una flotta specifica. Con le code, puoi aggiungere flotte con una nuova costruzione e rimuovere quelle vecchie se necessario. Per ulteriori informazioni, consulta Gestione del posizionamento delle sessioni di gioco con Amazon GameLift Servers code.
- Usa gli alias per trasferire i giocatori a una nuova build di gioco. Quando invii richieste di sessioni di gioco a Amazon GameLift Servers, specifica un alias della flotta anziché un ID della flotta. Per ulteriori informazioni, consulta Crea un Amazon GameLift Servers alias.
- Configurato per lo sviluppo iterativo. Durante lo sviluppo del gioco, esplora le opzioni per configurare un ambiente di test ospitato che supporti lo sviluppo iterativo rapido. Per informazioni, consulta Configurato per lo sviluppo iterativo con Amazon GameLift Servers Ovunque.

Prova queste risorse del Amazon GameLift Servers Toolkit su Github:

Strumento di aggiornamento rapido (solo per lo sviluppo)

> Questo strumento ti aiuta a modificare le build dei server di gioco già distribuite sui computer di una EC2 flotta gestita, facendoti risparmiare tempo durante le iterazioni di sviluppo rapide. Lo

strumento offre diverse opzioni: puoi sostituire l'intera build del gioco o modificare file specifici e puoi gestire il riavvio dei processi del server di gioco dopo gli aggiornamenti. Puoi anche usarlo per aggiornare tutti i computer di una flotta o scegliere come target singoli computer.

Visita il Amazon GameLift Servers Repo Toolkit in Github per scaricare lo strumento di aggiornamento rapido della build in Github e scoprire di più su come usarlo.

Script di esempio per la distribuzione in produzione

Questo script illustra come automatizzare il processo di aggiornamento delle build di server di gioco distribuite su flotte gestite EC2 in produzione. Per utilizzare questo script, Amazon GameLift Servers la soluzione di hosting deve utilizzare alias per astrarre la flotta IDs. Lo script di esempio gestisce i seguenti passaggi di base: caricare una build aggiornata, creare una nuova build e distribuirla in una nuova flotta, reindirizzare il traffico dei giocatori da una flotta esistente a quella nuova ed eliminare la vecchia flotta. Personalizza lo script di esempio per soddisfare i tuoi requisiti di implementazione specifici.

Visita il Amazon GameLift Servers Repo Toolkit in Github per scaricare lo script di esempio per la distribuzione di produzione in Github e scoprire di più su come usarlo.

Crea un'immagine del contenitore per Amazon GameLift Servers

Questo argomento descrive come creare un'immagine del contenitore con il software del server di gioco con Amazon GameLift Servers cui utilizzarla. L'immagine del contenitore del server di gioco include l'eseguibile del server di gioco e tutte le dipendenze necessarie per l'esecuzione. Le immagini dei container del server di gioco vengono utilizzate con una soluzione di hosting di container Amazon GameLift Servers gestiti. Per i dettagli sulla creazione della soluzione completa, consulta:

- Roadmap di sviluppo per l'hosting con contenitori Amazon GameLift Servers gestiti
- Come funzionano i contenitori in Amazon GameLift Servers

Completa le seguenti attività per preparare l'immagine del container del server di gioco per la distribuzione in una flotta di Amazon GameLift Servers container. Prima di iniziare queste attività, completa l'integrazione del codice del server di gioco con l'SDK del Amazon GameLift Servers server.

Argomenti

- Crea un'immagine del contenitore del server di gioco
- Invia l'immagine di un contenitore ad Amazon ECR

Crea un'immagine del contenitore del server di gioco

Segui queste istruzioni mentre lavori su una piattaforma basata su Linux o usi Windows Subsystem for Linux (WSL) con Docker installato.

Per creare un'immagine del contenitore del server di gioco

1. Prepara una directory di lavoro con il software del tuo server di gioco. Su un computer locale, crea una directory di lavoro per organizzare i file per il contenitore del tuo server di gioco. L'immagine del contenitore utilizza questa struttura di file quando distribuisce il contenitore su Amazon GameLift Servers risorse per l'hosting. Per esempio:

   ```
   [~/]$ mkdir -p work/glc/gamebuild && cd work && find .
   .
   ./glc
   ./glc/gamebuild
   ```

   > **Note**
   >
   > Se stai provando questa funzionalità e non hai ancora un server di gioco funzionante, prova il nostro server di gioco di esempio SimpleServer, disponibile su GitHub.

2. Crea un nuovo Dockerfile utilizzando il modello fornito.
3. Segui le istruzioni nel modello Dockerfile per aggiornarlo per uso personale.
   - Aggiorna l'immagine di base secondo necessità.
   - Imposta le variabili di ambiente per la build del tuo server di gioco.
4. Crea l'immagine del contenitore. Esegui il`docker build`, specificando il nome del tuo repository. Per esempio:

   ```
   [~/work/glc]$ docker build -t <local repository name>:<optional tag> .
   ```

   È possibile visualizzare i repository e le immagini IDs utilizzando il `docker images` comando, come illustrato in questo esempio:

Modello Dockerfile per l'immagine del contenitore di un server di gioco

Questo modello contiene le istruzioni minime di cui un contenitore di server di gioco ha bisogno per essere utilizzabile in una Amazon GameLift Servers flotta. Modifica il contenuto in base alle esigenze del tuo server di gioco.

```
# Base image
# ----------
  # Add the base image that you want to use,
  # Make sure to use an image with the same architecture as the
  # Instance type you are planning to use on your fleets.
FROM public.ecr.aws/amazonlinux/amazonlinux
  #
# Game build directory
# --------------------
  # Add your gamebuild directory to the env variable below.
  # The game build provided here needs to be integrated with server sdk for Amazon
 GameLift Servers.
ENV GAME_BUILD_DIRECTORY="<ADD_GAME_BUILD_DIRECTORY>" \
  #
# Game executable and launch parameters
# ---------------
  # Add the relative path to your executable in the 'GAME_EXECUTABLE' env variable
 below.
  # The game build provided over here needs to be integrated with server sdk for Amazon
 GameLift Servers.
  # This template assumes that the executable path is relative to the game build
 directory.
  # Add any launch parameters to pass into your executable in the 'LAUNCH_PARAMS' env
 variable below.
  # Add 'HOME_DIR' to identify where the game executable and logs exist.
GAME_EXECUTABLE="<ADD NAME OF EXECUTABLE WITHIN THE GAME DIRECTORY>" \
LAUNCH_PARAMS=<ADD LAUNCH PARAMETERS> \
HOME_DIR="/local/game" \


# Install dependencies as necessary
RUN yum install -y shadow-utils

RUN mkdir -p $HOME_DIR
COPY ./$GAME_BUILD_DIRECTORY/ $HOME_DIR

# Change directory to home
```

```
WORKDIR $HOME_DIR

# Set up for 'gamelift' user
RUN useradd -m gamelift && \
  echo "gamelift ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers && \
  chown -R gamelift:gamelift $HOME_DIR

# Add permissions to game build
RUN chmod +x ./$GAME_EXECUTABLE

USER gamelift

# Check directory before starting the container
RUN ls -lhrt .

# Check path before starting the container
RUN echo $PATH

# Start the game build
ENTRYPOINT ["/bin/sh", "-c", "./$GAME_EXECUTABLE", "$LAUNCH_PARAMS"]
```

Invia l'immagine di un contenitore ad Amazon ECR

Dopo aver creato un'immagine del contenitore per la distribuzioneAmazon GameLift Servers, archivia l'immagine in un repository pubblico o privato in Amazon ECR. A questo repository viene assegnato un valore URI, che viene Amazon GameLift Servers utilizzato per scattare un'istantanea dell'immagine per la distribuzione in una flotta di container.

> **Note**
>
> Se non disponi ancora di un repository privato Amazon ECR, creane uno.

Per inviare l'immagine del contenitore ad Amazon ECR

1. Ottieni le tue credenziali Amazon ECR. Prima di inviare un'immagine del contenitore ad Amazon ECR, acquisisci AWS le tue credenziali in formato temporaneo e forniscile a Docker. Docker necessita di queste credenziali per accedere.

   ```
   [~/work/glc]$ aws ecr get-login-password --region us-west-2 | docker login --
   username AWS --password-stdin aws_account_id.dkr.ecr.us-west-2.amazonaws.com
   ```

```
WARNING! Your password will be stored unencrypted in
/home/user-name/.docker/config.json.
Configure a credential helper to remove this warning.
See https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

2. Copia l'URI del repository privato Amazon ECR che desideri utilizzare.
3. Applica un tag Amazon ECR all'immagine del contenitore.

   Example

   ```
   [~/work/glc]$ docker tag <IMAGE ID from above> <Amazon ECR private repository
    URI>:<optional tag>
   ```

4. Invia l'immagine del contenitore ad Amazon ECR

   Example

   ```
   [~/work/glc]$ docker image push  <Amazon ECR private repository URI>
   ```

Implementa uno script per Amazon GameLift ServersRealtime

Quando sei pronto Amazon GameLift Servers Realtime per la distribuzione del gioco, carica i file di script Realtime del server completi su. Amazon GameLift Servers A tale scopo, crea una risorsa di Amazon GameLift Servers script e specifica la posizione dei file di script. È inoltre possibile aggiornare i file di script del server già distribuiti caricando nuovi file per una risorsa di script esistente.

Quando create una nuova risorsa di script, le Amazon GameLift Servers assegna un ID di script univoco (ad esempio`script-1111aaaa-22bb-33cc-44dd-5555eeee66ff`) e carica una copia dei file di script. Il tempo di caricamento dipende dalla dimensione dei file di script e dalla velocità di connessione.

Dopo aver creato la risorsa di script, Amazon GameLift Servers distribuisce lo script con una nuova Amazon GameLift Servers Realtime flotta. Amazon GameLift Serversinstalla lo script del server su ogni istanza del parco istanze, inserendo i file di script. `/local/game`

Per risolvere i problemi di attivazione del parco veicoli relativi allo script del server, consulta. Debug dei problemi del parco istanze di Amazon GameLift Servers

File di script Package

Lo script del server può includere uno o più file combinati in un unico file.zip da caricare. Il file.zip deve contenere tutti i file necessari per l'esecuzione dello script.

Puoi archiviare i file di script compressi in una directory di file locale o in un bucket Amazon Simple Storage Service (Amazon S3).

Carica file di script da una directory locale

Se i file di script sono archiviati localmente, puoi caricarli Amazon GameLift Servers da lì. Per creare la risorsa script, usa la Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI).

GameLift Server Amazon console

Per creare una risorsa di script

1. Apri la Amazon GameLift Servers console.
2. Nel pannello di navigazione, scegli Hosting, Script.
3. Nella pagina Script, scegli Crea script.
4. Nella pagina Crea script, in Impostazioni script, procedi come segue:
    a. Per Nome, immettete il nome dello script.
    b. (Facoltativo) In Versione, immettere le informazioni sulla versione. Poiché è possibile aggiornare il contenuto di uno script, i dati sulla versione possono essere utili per tenere traccia degli aggiornamenti.
    c. Per Script source, scegli Carica un file.zip.
    d. Per i file di script, scegli Scegli file, cerca il file.zip che contiene lo script, quindi scegli quel file.
5. (Facoltativo) In Tag, aggiungete i tag allo script inserendo le coppie Chiave e Valore.
6. Scegli Create (Crea).

    Amazon GameLift Serversassegna un ID al nuovo script e carica il file.zip designato. È possibile visualizzare il nuovo script, incluso il relativo stato, nella pagina Script.

AWS CLI

Utilizza il comando `create-script` AWS CLI per definire il nuovo script del server e caricare i file dello script.

Per creare una risorsa di script

1. Inserisci il file.zip in una directory in cui puoi usare. AWS CLI
2. Apri una finestra della riga di comando e passa alla directory in cui hai inserito il file.zip.
3. Immettete il create-script comando e i parametri seguenti. Per il --zip-file parametro, assicuratevi di aggiungere la stringa `fileb://` al nome del file.zip. Identifica il file come binario in modo da Amazon GameLift Servers elaborare il contenuto compresso.

```
aws gamelift create-script \
    --name user-defined name of script \
    --script-version user-defined version info \
    --zip-file fileb://name of zip file \
    --region region name
```

Esempio

```
aws gamelift create-script \
    --name "My_Realtime_Server_Script_1" \
    --script-version "1.0.0" \
    --zip-file fileb://myrealtime_script_1.0.0.zip \
    --region us-west-2
```

In risposta alla richiesta, Amazon GameLift Servers restituisce il nuovo oggetto script.

4. Per visualizzare il nuovo script, chiamadescribe-script.

Caricare file di script da Amazon S3

Puoi archiviare i tuoi file di script in un bucket Amazon S3 e caricarli da Amazon GameLift Servers lì. Quando crei lo script, specifichi la posizione del bucket S3 e Amazon GameLift Servers recuperi i file di script da Amazon S3.

Per creare una risorsa di script

1. Archivia i tuoi file di script in un bucket S3. Crea un file.zip contenente i file di script del server e caricalo in un bucket S3 in un ambiente sotto il tuo controllo. Account AWS Prendi nota dell'URI dell'oggetto: ne hai bisogno quando crei uno script. Amazon GameLift Servers

   > **Note**
   >
   > Amazon GameLift Serversnon supporta il caricamento da bucket S3 con nomi che contengono un punto (.).

2. Concedi ad Amazon GameLift Servers l'accesso ai file dello script. Per creare un ruolo AWS Identity and Access Management (IAM) che Amazon GameLift Servers consenta di accedere al bucket S3 contenente lo script del server, segui le istruzioni riportate in. Configurare un ruolo di servizio IAM per Amazon GameLift Servers Dopo aver creato il nuovo ruolo, prendi nota del suo nome, che ti serve per creare uno script.
3. Creare uno script. Usa la Amazon GameLift Servers console o il AWS CLI per creare un nuovo record di script. Per effettuare questa richiesta, è necessario disporre dell'`PassRole`autorizzazione IAM, come descritto inEsempi di autorizzazioni IAM per Amazon GameLift Servers.

GameLift Server Amazon console

1. Nella Amazon GameLift Serversconsole, nel pannello di navigazione, scegli Hosting, Scripts.
2. Nella pagina Script, scegli Crea script.
3. Nella pagina Crea script, in Impostazioni script, procedi come segue:
   a. Per Nome, immettete il nome dello script.
   b. (Facoltativo) In Versione, immettere le informazioni sulla versione. Poiché è possibile aggiornare il contenuto di uno script, i dati sulla versione possono essere utili per tenere traccia degli aggiornamenti.
   c. Per l'origine dello script, scegli l'URI di Amazon S3.
   d. Inserisci l'URI S3 dell'oggetto script che hai caricato su Amazon S3, quindi scegli la versione dell'oggetto. Se non ricordi l'URI e la versione dell'oggetto di Amazon S3, scegli Browse S3, quindi cerca l'oggetto script.
4. (Facoltativo) In Tag, aggiungi tag allo script inserendo le coppie Chiave e Valore.

5. Scegli Create (Crea).

   Amazon GameLift Serversassegna un ID al nuovo script e carica il file.zip designato. È possibile visualizzare il nuovo script, incluso il relativo stato, nella pagina Script.

AWS CLI

Utilizza il comando `create-script` AWS CLI per definire il nuovo script del server e caricare i file dello script.

1. Aprire una finestra della riga di comando e passare a una directory in cui è possibile utilizzare. AWS CLI
2. Immettete il create-script comando e i parametri seguenti. Il --storage-location parametro specifica la posizione del bucket Amazon S3 dei file di script.

```
aws gamelift create-script \
    --name [user-defined name of script] \
    --script-version [user-defined version info] \
    --storage-location "Bucket"=S3 bucket name,"Key"=name of zip file in S3
 bucket,"RoleArn"=Access role ARN \
    --region region name
```

   Esempio

```
aws gamelift create-script \
    --name "My_Realtime_Server_Script_1" \
    --script-version "1.0.0" \
    --storage-location "Bucket"="gamelift-
script","Key"="myrealtime_script_1.0.0.zip","RoleArn"="arn:aws:iam::123456789012:role/
S3Access" \
    --region us-west-2
```

   In risposta alla tua richiesta, Amazon GameLift Servers restituisce il nuovo oggetto script.

3. Per visualizzare il nuovo script, chiama `describe-script`.

Aggiorna i file di Amazon GameLift ServersRealtime script

È possibile aggiornare i metadati per una risorsa di script utilizzando la Amazon GameLift Servers console o il `update-script` AWS CLI comando.

È inoltre possibile aggiornare il contenuto dello script per una risorsa di script. Amazon GameLift Serversdistribuisce il contenuto dello script in tutte le istanze del parco istanze che utilizzano la risorsa di script aggiornata. Quando lo script aggiornato viene distribuito, le istanze lo utilizzano per avviare nuove sessioni di gioco. Le sessioni di gioco già in esecuzione al momento dell'aggiornamento non utilizzano lo script aggiornato.

Per aggiornare i file di script

- Per i file di script archiviati localmente, per caricare il file.zip di script aggiornato, usa la Amazon GameLift Servers console o il update-script comando.
- Per i file di script archiviati in un bucket Amazon S3, carica i file di script aggiornati nel bucket S3. Amazon GameLift Serversverifica periodicamente la presenza di file di script aggiornati e li recupera direttamente dal bucket S3.

Configurazione di una flotta di hosting con Amazon GameLift Servers

In questa sezione troverai informazioni sulla progettazione, la costruzione e la manutenzione di Amazon GameLift Servers flotte per ospitare i tuoi server di gioco. Scopri Amazon GameLift Serversopzioni di hosting di più sulle soluzioni di hosting Amazon GameLift Servers offerte, comprese quelle che utilizzano EC2 flotte gestite, flotte Anywhere autogestite per l'hardware locale e una soluzione ibrida che utilizza entrambe.

Argomenti

Caratteristiche della flotta

Una Amazon GameLift Servers flotta è una raccolta di risorse informatiche che gestiscono i server di gioco e ospitano sessioni di gioco per i giocatori. Le flotte possono variare in base al tipo di risorse di elaborazione utilizzate e al modo in cui viene gestita la flotta. Le dimensioni di una flotta, ovvero il numero di sessioni di gioco e di giocatori che può supportare, dipendono dal numero di risorse di elaborazione che le fornisci. Tutte le flotte hanno le seguenti caratteristicheAmazon GameLift Servers:

- I processi del server di gioco eseguiti su tutte le flotte sono integrati con l'SDK del server Amazon GameLift Servers e comunicano con il Amazon GameLift Servers servizio allo stesso modo. I server di gioco segnalano la loro disponibilità a ospitare sessioni di gioco e giocatori, rispondono alle richieste di avvio o interruzione delle sessioni di gioco e altre interazioni.
- Amazon GameLift Serversgestisce il posizionamento delle sessioni di gioco per tutte le flotte allo stesso modo. Amazon GameLift Serverstiene traccia dello stato del server di gioco di una flotta e sceglie tra i server di gioco disponibili per ospitare una nuova sessione di gioco. Questo processo

viene utilizzato se il gioco colloca le sessioni di gioco su una singola flotta o utilizza una coda di sessioni di gioco per bilanciare l'hosting tra più flotte. Con una coda, puoi anche personalizzare le decisioni di posizionamento tenendo conto di fattori come il costo delle risorse e la latenza.

- Tutte le flotte supportano l'uso di un FlexMatch matchmaker in collaborazione con una coda per il posizionamento delle sessioni di gioco. Il Amazon GameLift Servers servizio riceve le richieste di partite dei giocatori, forma le partite e le passa alla coda delle sessioni di gioco per trovare i server di gioco disponibili.
- Amazon GameLift Serversraccoglie un'ampia gamma di metriche sulla flotta. Queste includono metriche di stato per computer e processi del server, nonché metriche di utilizzo per le sessioni di gioco e l'attività dei giocatori. Consulta l'elenco completo delle metriche disponibili all'indirizzo. Monitora Amazon GameLift Servers con Amazon CloudWatch

Come funziona Amazon GameLift Servers la creazione della flotta

Quando richiedi una nuova flotta, Amazon GameLift Servers avvia un flusso di lavoro per creare la risorsa del parco veicoli. Al termine di ogni fase del flusso di lavoro, Amazon GameLift Servers aggiorna lo stato della flotta ed emette una serie di eventi per comunicare i progressi verso la creazione della flotta.

Amazon GameLift Serversutilizza due tipi di eventi. Gli eventi di transizione dello stato della flotta segnano quando lo stato della flotta cambia. Gli eventi di creazione della flotta forniscono indicatori aggiuntivi per risolvere i problemi di creazione del debug. Puoi tenere traccia di tutti gli eventi utilizzando la Amazon GameLift Servers console o chiamando l'operazione API. Amazon GameLift Servers DescribeFleetEvents Puoi anche monitorare lo stato della flotta e della posizione utilizzando DescribeFleetAttributeso DescribeFleetLocationAttributes.

Amazon GameLift Servers EC2 flotte gestite

Amazon GameLift Servers EC2 le flotte gestite offrono risorse basate sul cloud per l'hosting di produzione con Amazon GameLift Servers. Con una flotta gestita, ottieni la flessibilità, la sicurezza e l'affidabilità di Cloud AWS risorse ulteriormente ottimizzate per l'hosting di giochi multiplayer. Il Amazon GameLift Servers il servizio fornisce solidi strumenti di gestione degli host.

Una EC2 flotta gestita è un insieme di computer virtuali che Amazon GameLift Servers possiede e opera per tuo conto e in base alle tue scelte di configurazione. I computer sono istanze di Amazon Elastic Compute Cloud EC2 (Amazon) che si trovano fisicamente nelle nostre Local Regioni AWS

Zones. Quando crei una flotta, scegli un tipo di EC2 istanza per i tuoi computer in base a potenza di calcolo, memoria, storage, funzionalità di rete e altri fattori.

Con una EC2 flotta gestita, carichi la build del tuo server di gioco su Amazon GameLift Servers. Quando crei una flotta, il servizio distribuisce la build sui computer della flotta e avvia i processi del server di gioco. Ogni processo del server di gioco avviato stabilisce una connessione al Amazon GameLift Servers servizio e segnala la disponibilità a ospitare una sessione di gioco.

Oltre al dispiegamento della flotta, Amazon GameLift Servers gestisce le seguenti attività di gestione dell'host in modo da non dover:

- Tiene traccia dello stato di tutti i computer del parco computer e sostituisce i computer obsoleti o non integri.
- Gestisce l'autenticazione per la comunicazione tra i processi del server e il Amazon GameLift Servers servizio.
- Avvia e arresta automaticamente i processi del server di gioco su ogni computer, in base alla configurazione di runtime.
- Offre strumenti di auto-scaling per regolare la capacità della flotta in modo dinamico per soddisfare la domanda dei giocatori.
- Riporta le metriche delle prestazioni per le istanze del parco istanze. EC2

Consulta questi argomenti su come configurare e mantenere flotte gestite EC2 :

- Roadmap di sviluppo per l'hosting con managed Amazon GameLift Servers EC2
- Crea una EC2 flotta Amazon GameLift Servers gestita
- Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers
- Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite
- Aggiornamento di una configurazione del Amazon GameLift Servers parco veicoli

Flusso di lavoro gestito EC2 per la creazione di

Per flotte gestite, Amazon GameLift Servers configura la risorsa del parco macchine e distribuisce anche una serie di risorse di calcolo con il software del server di gioco installato e in esecuzione. Quando il flusso di lavoro di creazione è completo e ha esito positivo, la flotta dispone di un' EC2 istanza attiva nella regione di origine della flotta e una in ciascuna delle località remote della flotta. Tutte le istanze dispongono di giochi pronti per ospitare sessioni di gioco.

1. Amazon GameLift Servers crea la risorsa del parco veicoli nella regione di origine della flotta e imposta la capacità desiderata in ogni sede su una (1) istanza. Lo stato della flotta e dell'ubicazione è impostato su Nuovo.
2. Amazon GameLift Servers inizia a scrivere gli eventi nel registro degli eventi della flotta.
3. Amazon GameLift Servers imposta lo stato della flotta su Downloading e inizia a preparare il software del server di gioco per l'implementazione.
   a. Ottiene la build del server di gioco caricata ed estrae i file compressi.
   b. Esegue gli script di installazione, se forniti.
   c. Imposta lo stato del parco macchine su Convalida e inizia a verificare che non si siano verificati errori durante il download e l'installazione dei file di build.
4. Amazon GameLift Servers imposta lo stato del parco macchine su Building, configura l'hardware del parco macchine e alloca un' EC2 istanza per ogni istanza del parco macchine.
5. Amazon GameLift Servers imposta lo stato della flotta su Attivazione. Avvia un processo del server di gioco su ogni istanza (in base alle istruzioni di runtime della flotta) e verifica la connettività tra la build e il Amazon GameLift Servers servizio.
6. Quando i processi del server di gioco su ciascuna istanza stabiliscono una connessione e segnalano la disponibilità a ospitare sessioni di gioco, Amazon GameLift Servers imposta gli stati della flotta e della posizione su Attivo. A questo punto, la flotta è considerata pronta per ospitare sessioni di gioco.

Crea una EC2 flotta Amazon GameLift Servers gestita

Questo argomento descrive come creare un EC2 parco veicoli Amazon GameLift Servers gestito. Le flotte gestite utilizzano istanze di calcolo Amazon Elastic Compute Cloud EC2 (Amazon) ottimizzate per l'hosting di giochi multiplayer. Puoi creare flotte gestite che distribuiscono i computer a livello globale verso le Local Regioni AWS Zones supportate da. Amazon GameLift Servers

Quando crei una nuova EC2 flotta gestita, il processo di creazione della flotta inizia immediatamente. Una flotta gestita passa attraverso diverse fasi: Amazon GameLift Servers prepara la build del server di gioco, distribuisce EC2 le istanze con la build installata e avvia i server di gioco su ciascuna istanza. Puoi monitorare lo stato di una flotta nella console o utilizzando uring (). AWS Command Line Interface AWS CLI Una flotta è pronta per ospitare sessioni di gioco quando il suo stato viene raggiunto`ACTIVE`. Per ulteriori informazioni sulla creazione gestita della flotta, consulta i seguenti argomenti:

- Come funziona Amazon GameLift Servers la creazione della flotta
- Debug dei problemi del parco istanze di Amazon GameLift Servers

Per creare una EC2 flotta gestita

Usa la Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI) per creare una EC2 flotta gestita.

Console

Nella Amazon GameLift Serversconsole, utilizza il riquadro di navigazione per aprire la pagina Fleets. Scegli Crea flotta per avviare il flusso di lavoro di creazione della flotta.

Passaggio 1 Scegli il tipo di elaborazione

Seleziona l' EC2opzione Gestito e scegli Avanti.

Fase 2 Definire i dettagli della flotta

In questo passaggio, specifica alcune impostazioni a livello di flotta.

> Per una configurazione minima del parco veicoli:
> - Fornisci un nome per la flotta.
> - Scegli un tipo binario e specifica una build o uno script caricato.
> - Salta le sezioni relative a dettagli e tag aggiuntivi.

1. Compila la sezione dei dettagli della flotta:
   a. Inserisci un nome per la flotta. Ti consigliamo di utilizzare uno schema di denominazione della flotta che faciliti l'identificazione dei tipi di flotta durante la visualizzazione degli elenchi di flotte.
   b. Fornisci una breve descrizione della flotta.
   c. Per il tipo binario, seleziona Build per indicare che stai implementando una build di server di gioco personalizzata, oppure seleziona o Script se stai distribuendo su questa Amazon GameLift Servers Realtime flotta. Seleziona una build o uno script caricato dall'elenco a discesa.
2. (Facoltativo) Imposta dettagli aggiuntivi in base alle esigenze.

a. Se l'eseguibile del server di gioco deve accedere ad altre AWS risorse del tuo account, specifica un ruolo di istanza IAM con le autorizzazioni necessarie. Per ulteriori informazioni, incluso come autorizzare altre applicazioni lato server (come l' CloudWatch agente), consulta. Comunica con altre AWS risorse delle tue flotte Questa impostazione non può essere modificata dopo aver creato la flotta.

   È necessario creare il ruolo prima di creare una flotta che lo utilizzi. Inoltre, per creare una flotta con un ruolo di istanza, AWS l'utente deve disporre dell'`PassRole`autorizzazione IAM (vediEsempi di autorizzazioni IAM per Amazon GameLift Servers).

b. Attiva l'opzione Genera un certificato TLS per configurare l'autenticazione e la crittografia per il gioco. I client di gioco utilizzano questo certificato per autenticare un server di gioco durante la connessione e crittografare tutte le comunicazioni client/server. Per ogni istanza in un parco istanze abilitato per TLS, Amazon GameLift Servers crea anche una nuova voce DNS con il certificato. Questa impostazione non può essere modificata dopo aver creato la flotta.

c. Se desideri combinare i dati metrici per questo parco veicoli e altri, specifica il nome di un gruppo di metriche. Utilizza lo stesso nome di gruppo metrico per tutte le flotte che desideri combinare. Visualizza le metriche per il gruppo di metriche per vedere i dati aggregati.

3. (Facoltativo) Aggiungi tag alla risorsa del parco veicoli. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Assegna tag alle AWS risorse che desideri classificare in modi utili, ad esempio per scopo, proprietario o ambiente. Scegli Aggiungi nuovo tag per ogni tag che desideri aggiungere.
4. Scegli Avanti per continuare il flusso di lavoro.

Fase 3 Definire i dettagli dell'istanza

In questo passaggio, specifica il tipo di risorse di hosting da utilizzare e dove desideri distribuirle. Scegliendo più sedi, puoi distribuire il server di gioco in una posizione geografica più ampia, in modo da avvicinarli ai giocatori e ridurre al minimo la latenza. Non tutti i tipi di EC2 istanze sono disponibili in tutte le località.

> Per una configurazione minima del parco veicoli:
>
> - Non aggiungere postazioni remote.

- Imposta il tipo di flotta impostato su «On-Demand». Le flotte Spot richiedono un lavoro di configurazione aggiuntivo.
- Imposta il tipo di istanza su «c5.large». Questo tipo di istanza comunemente usato è disponibile in tutti. Regioni AWS

1. Nella distribuzione dell'istanza, specifica l'ubicazione e il tipo della flotta.

   a. Seleziona una o più sedi aggiuntive in cui desideri distribuire le istanze del parco istanze. Queste postazioni remote vengono aggiunte alla sede principale della flotta (che è preselezionata), che è la posizione Regione AWS in cui stai creando questa flotta. È possibile selezionare postazioni remote tra tutte Regioni AWS le Local Zones Amazon GameLift Servers supportate.

      Per ulteriori informazioni sulle località supportate, incluso come utilizzare quelle Regione AWS che non sono abilitate Amazon GameLift Serverssedi di assistenza per impostazione predefinita, consulta Informazioni sull'hosting gestito. Consulta anche Amazon GameLift Servers le quote relative alle sedi per flotta.

   b. Scegli di utilizzare istanze On-demand o Spot per questo parco istanze. Per ulteriori informazioni sui tipi di parco veicoli, consulta. Istanze On-Demand e istanze Spot

2. Scegli una configurazione Amazon EC2 Instance che soddisfi le tue esigenze e sia disponibile in tutte le località selezionate. Questo elenco viene filtrato in base alla posizione corrente e alle selezioni del tipo di flotta. Puoi filtrarlo ulteriormente in base ad altri fattori, come il tipo di istanza, la famiglia e l'architettura. Dopo aver creato il parco istanze, non puoi modificare il tipo di istanza.

   Alcune sedi hanno opzioni di tipo di istanza limitate. Se il tipo di istanza preferito non è disponibile per tutte le sedi, scegli il valore di disponibilità della sede per visualizzare tutti i dettagli. Per ospitare tutte le sedi, potrebbe essere necessario creare flotte separate con diversi tipi di istanze.

   Per ulteriori informazioni sulla scelta del tipo di istanza, consultaTipi di istanza.
   Per ulteriori informazioni sulle architetture Amazon EC2 Arm, consulta Processore AWS Graviton e tipi di istanze Amazon EC2. Per un elenco completo dei tipi di istanze supportati daAmazon GameLift Servers, consulta il riferimento API per EC2InstanceType(). `CreateFleet()`

> **Note**
>
> Le istanze Graviton Arm richiedono un Amazon GameLift Servers server costruito su sistema operativo Linux. Server SDK 5.1.1 o versione successiva è richiesto per C++ e C#. Server SDK 5.0 o versione successiva è richiesto per Go. Queste istanze non forniscono out-of-the-box supporto per l'installazione di Mono su Amazon Linux 2023 (AL2023) o Amazon Linux 2 (). AL2

3. Scegli Avanti per continuare il flusso di lavoro.

Fase 4: Configurare il runtime

In questo passaggio, descrivi come desideri che ogni istanza del parco istanze esegua il software del server di gioco. Definisci una riga di processo del server distinta per ogni eseguibile da eseguire su un'istanza e decidi quanti processi server eseguire contemporaneamente. Apri le porte su ogni istanza per consentire ai giocatori di connettersi direttamente ai server di gioco. Puoi aggiornare le impostazioni di queste flotte in qualsiasi momento.

> Per una configurazione minima del parco veicoli:
>
> - Definisci un singolo elemento della riga di processo del server per il file eseguibile del server di gioco. Se il server di gioco richiede l'esecuzione di altri processi, crea anche una definizione per ciascuno di questi.
> - Utilizza il numero predefinito di processi simultanei (1) per ogni elemento della riga.
> - Salta le impostazioni di attivazione della sessione di gioco.
> - Specificare un numero di porta singolo.
> - Salta le impostazioni delle risorse della sessione di gioco.

1. Crea una configurazione Runtime per istruzioni Amazon GameLift Servers su come eseguire i processi del server su ogni istanza del parco istanze. Puoi modificare la configurazione di runtime di una flotta in qualsiasi momento dopo l'implementazione.

   a. Inserisci il percorso di lancio di un file eseguibile nella tua build. Nelle istanze di Windows, gli eseguibili dei server di gioco vengono creati in base al percorso. `C:`

`\game` Nelle istanze Linux, i server di gioco sono progettati per. `/local/game` Esempi: **`C:\game\MyGame\server.exe`**`/local/game/MyGame/server.exe`, o**`MyRealtimeLaunchScript.js`**.

b. Inserisci i parametri di avvio opzionali da passare al file eseguibile del gioco. Esempio: **`+sv_port 33435 +start_lobby`**.

c. Specificate il numero di processi simultanei da eseguire su ogni istanza. Per un server di gioco eseguibile, ogni processo può ospitare una sessione di gioco, quindi i processi simultanei determinano il numero di sessioni di gioco che l'istanza può ospitare contemporaneamente.

Controlla le Amazon GameLift Servers quote sui processi del server per istanza. Queste quote si applicano al totale dei processi simultanei per tutte le configurazioni. Se configuri la flotta in modo da superarle, la flotta non può attivarsi.

2. Usa le impostazioni predefinite per l'attivazione della sessione di gioco o personalizzale per il tuo gioco. Se la configurazione di runtime richiede più processi simultanei del server di gioco per istanza, queste impostazioni determinano la velocità di avvio delle nuove sessioni di gioco.

a. Imposta l'attivazione massima della sessione di gioco simultanea per limitare il numero di server di gioco su un'istanza che stanno preparando una nuova sessione di gioco. Questa impostazione è utile quando l'avvio di più nuove sessioni di gioco richiede molte risorse e potrebbe influire sulle prestazioni di altre sessioni di gioco in corso.

b. Imposta il nuovo timeout di attivazione in modo da indicare il tempo massimo che una nuova sessione di gioco deve impiegare per completare l'attivazione e segnalare che è pronta per ospitare i giocatori. Amazon GameLift Serversinterrompe l'attivazione di una sessione di gioco se supera questo valore.

3. Apri le impostazioni delle EC2 porte per consentire al traffico in entrata di accedere ai processi del server del parco macchine. Queste impostazioni non sono necessarie per creare una flotta, ma è necessario configurarle prima che i giocatori possano connettersi alle sessioni di gioco della flotta.

Per ogni impostazione di porta, scegli il tipo di protocollo di trasferimento dati da utilizzare per la comunicazione tra il client di gioco e il server di gioco. Fornisci un intervallo di porte (in formatonnnnn`[-nnnnn]`) e un intervallo di indirizzi IP utilizzando la notazione CIDR (ad esempio, **`0.0.0.0/0`** che consente l'accesso a chiunque).

Se devi impostare più intervalli non consecutivi, crea più impostazioni di porta.

4. Specificate le impostazioni opzionali delle risorse della sessione di gioco. Puoi aggiornare queste impostazioni in qualsiasi momento dopo la distribuzione.

   a. Attiva o disattiva la politica di protezione di Game Scaling per tutte le istanze del parco istanze. Durante un evento di ridimensionamento, non Amazon GameLift Servers interromperà le istanze protette del parco istanze che ospitano sessioni di gioco attive.

   b. Imposta un limite massimo per la creazione di risorse se desideri limitare il numero di sessioni di gioco che un giocatore può creare durante un periodo di tempo specificato.

5. Scegli Avanti per continuare il flusso di lavoro.

Passaggio 5: rivedi e crea

Controlla le impostazioni prima di creare la flotta. Sebbene alcune impostazioni possano essere aggiornate in un secondo momento (vediAggiornamento di una configurazione del Amazon GameLift Servers parco veicoli), le modifiche alle seguenti impostazioni non sono consentite dopo la creazione del parco veicoli:

- Tipo di elaborazione: non puoi convertire una EC2 flotta gestita in una flotta Anywhere.
- Compilazione o script: per implementare un aggiornamento alla build o allo script del server di gioco, devi creare una nuova flotta.
- Opzioni aggiuntive, tra cui il ruolo dell'istanza e la generazione di certificati TLS.
- Dettagli dell'istanza, incluso il tipo di flotta (Spot o On-Demand) e il tipo di EC2 istanza.

Quando sei pronto per implementare la nuova flotta, scegli Crea. Amazon GameLift Serversavvia immediatamente il processo di attivazione della flotta, assegnando un ID univoco e impostando lo stato della flotta. `NEW` Tieni traccia dei progressi della flotta dalla pagina Flotte. Visualizza la pagina dei dettagli della flotta e vai alla scheda Eventi.

Puoi modificare la capacità di hosting di una flotta dopo che la flotta ha raggiunto lo stato ATTIVO. Amazon GameLift Serversinizialmente implementa una flotta con una singola istanza in ogni sede del parco macchine e tu modifichi la capacità aggiungendo istanze a ciascuna sede. Per ulteriori informazioni, consulta Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers.

AWS CLI

Usa il `create-fleet`comando per creare una flotta di tipi EC2 di calcolo. Amazon GameLift Serverscrea la risorsa fleet con l'impostazione predefinita corrente Regione AWS (oppure puoi aggiungere un tag --region per specificarne un altro Regione AWS).

Crea una flotta gestita minima

La seguente richiesta di esempio crea una nuova flotta con le impostazioni minime necessarie per implementare una flotta con server di gioco in esecuzione a cui i client di gioco possono connettersi. La nuova flotta presenta le seguenti caratteristiche:

- Specifica una build del server di gioco, che è stata caricata su Amazon GameLift Servers e in `READY` stato.
- Utilizza istanze On-Demand c5.large con un sistema operativo che corrisponde alla build del gioco selezionata.
- Imposta la sede della flotta `us-west-2` e distribuisce Regione AWS le istanze solo in quella regione.
- In base alla configurazione di runtime, ogni computer del parco computer esegue un processo del server di gioco, il che significa che ogni computer può ospitare solo una sessione di gioco alla volta. Il timeout di attivazione della sessione di gioco è impostato sul valore predefinito di 300 secondi e non c'è limite al numero di attivazioni simultanee.
- I giocatori possono connettersi ai server di gioco utilizzando l'impostazione di una singola porta di. `33435`
- Tutte le altre funzionalità sono disattivate o utilizzano le impostazioni predefinite.

```
aws gamelift create-fleet \
    --name MinimalFleet123 \
    --description "A basic test fleet" \
    --region us-west-2 \
    --ec2-instance-type c5.large \
    --fleet-type ON_DEMAND \
    --build-id build-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
    --runtime-configuration "ServerProcesses=[{LaunchPath=C:\game\Bin64.dedicated
\MultiplayerSampleProjectLauncher_Server.exe, ConcurrentExecutions=10}]" \
    --ec2-inbound-permissions
 "FromPort=33435,ToPort=33435,IpRange=0.0.0.0/0,Protocol=UDP"
```

Crea una flotta gestita completamente configurata

La seguente richiesta di esempio crea una flotta di produzione con impostazioni per tutte le funzionalità opzionali. La nuova flotta presenta le seguenti caratteristiche:

- Specifica una build del server di gioco, che è stata caricata su Amazon GameLift Servers e in `READY` stato.
- Utilizza le istanze On-Demand c5.large con il sistema operativo che corrisponde alla build di gioco selezionata.
- Imposta la sede del parco istanze `us-west-2` e distribuisce Regione AWS le istanze nella regione di origine e in una postazione remota. `sa-east-1`
- In base alla configurazione di runtime:
  - Ogni computer del parco computer esegue 10 processi del server di gioco con gli stessi parametri di avvio, il che significa che ogni computer può ospitare fino a 10 sessioni di gioco contemporaneamente.
  - Su ogni computer, possono essere attivate solo due sessioni di gioco contemporaneamente. Le sessioni di gioco attivate devono essere pronte a ospitare giocatori entro 300 secondi (5 minuti) o essere interrotte.
- I giocatori possono connettersi ai server di gioco utilizzando una porta compresa nel seguente intervallo. `33435 to 33535`
- Genera un certificato TLS per le comunicazioni crittografate tra client di gioco e server.
- In tutte le sessioni di gioco della flotta è attivata la protezione della sessione di gioco.
- I singoli giocatori sono limitati a creare tre nuove sessioni di gioco entro un periodo di 15 minuti.
- Le metriche per questa flotta sono incluse nel gruppo di metriche`AMERfleets`, che (per questo esempio) aggrega le metriche per un gruppo di flotte in Nord, Centro e Sud America.

```
aws gamelift create-fleet \
    --name ProdFleet123 \
    --description "A fully configured prod fleet" \
    --ec2-instance-type c5.large \
    --region us-west-2 \
    --locations "Location=sa-east-1" \
    --fleet-type ON_DEMAND \
    --build-id build-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
    --certificate-configuration "CertificateType=GENERATED" \
```

```
    --runtime-configuration "GameSessionActivationTimeoutSeconds=300,
 MaxConcurrentGameSessionActivations=2, ServerProcesses=[{LaunchPath=C:\game
\Bin64.dedicated\MultiplayerSampleProjectLauncher_Server.exe, Parameters=+sv_port
 33435 +start_lobby, ConcurrentExecutions=10}]" \
    --new-game-session-protection-policy "FullProtection" \
    --resource-creation-limit-policy "NewGameSessionsPerCreator=3,
 PolicyPeriodInMinutes=15" \
    --ec2-inbound-permissions
 "FromPort=33435,ToPort=33535,IpRange=0.0.0.0/0,Protocol=UDP" \
    --metric-groups  "AMERfleets"
```

Se la richiesta di creazione della flotta ha esito positivo, Amazon GameLift Servers restituisce un set di attributi della flotta che include le impostazioni di configurazione richieste e un nuovo ID della flotta. Amazon GameLift Serversavvia quindi il processo di attivazione della flotta e imposta lo stato della flotta e gli stati della posizione su Nuovo. Puoi monitorare lo stato del parco istanze e visualizzare altre informazioni tramite questi comandi dell'interfaccia CLI:

- describe-fleet-events
- describe-fleet-attributes
- describe-fleet-capacity
- describe-fleet-port-settings
- describe-fleet-utilization
- describe-runtime-configuration
- describe-fleet-location-attributes
- describe-fleet-location-capacity
- describe-fleet-location-utilization

È possibile modificarne la capacità e altre impostazioni di configurazione in base alle esigenze tramite questi comandi:

- update-fleet-attributes
- update-fleet-capacity
- update-fleet-port-settings
- update-runtime-configuration
- create-fleet-locations
- delete-fleet-locations

Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite

Gli argomenti di questa sezione descrivono alcune delle personalizzazioni che è possibile apportare quando si crea una soluzione di hosting con Amazon GameLift Servers flotte gestite EC2 . Forniscono indicazioni e best practice per creare e configurare una flotta per ospitare il gioco.

Le decisioni richieste includono:

- Dove dovresti distribuire le risorse di hosting? La latenza è un fattore importante nella scelta delle sedi del parco veicoli, ma i costi variano anche in base alla località.
- Quale tipo di EC2 istanza supporterà meglio il tuo gioco? Scegli tra i tipi di istanze disponibili in tutte le sedi del tuo parco macchine per utilizzare la migliore combinazione di architettura di elaborazione, memoria, archiviazione e capacità di rete.
- Di che dimensione del tipo di istanza hai bisogno? Scegli la dimensione del tipo di istanza in base ai requisiti di risorse (memoria e CPU) del software del server di gioco e ad altri fattori.
- La tua flotta dovrebbe utilizzare istanze On-Demand o Spot? Valuta se puoi trarre vantaggio dai prezzi Spot più bassi, visto come Amazon GameLift Servers protegge dalla possibilità di interruzioni delle sessioni di gioco.
- Come vuoi che funzioni il software del tuo server di gioco su ogni istanza del parco istanze? La configurazione di runtime lo dice Amazon GameLift Servers quale software server eseguire e come.

Argomenti

Scegli le risorse di elaborazione per una flotta gestita

Per distribuire i server di gioco e ospitare sessioni di gioco nel cloud, Amazon GameLift Servers fornisce flotte gestite che utilizzano risorse Amazon Elastic Compute Cloud EC2 (Amazon) chiamate istanze. Utilizza i seguenti argomenti per decidere quale tipo di EC2 istanze desideri utilizzare per la tua soluzione di hosting gestito e come configurarle per eseguire il software del server di gioco.

> **Note**
>
> Se prevedi di utilizzare risorse di hosting di tua proprietà, hardware locale o altro tipo di hosting basato sul cloud, prendi in considerazione le opzioni per l'hosting ibrido con Amazon GameLift Servers Ovunque. Per informazioni, consulta Configurazione di una flotta di hosting con Amazon GameLift Servers.

Argomenti

Ubicazione della flotta

Considera le aree geografiche in cui prevedi di installare i tuoi server di gioco. La disponibilità del tipo di istanza varia in base Regione AWS alla zona locale.

Per le flotte con più sedi, la disponibilità e le quote delle istanze dipendono dalla combinazione della regione di origine del parco veicoli e di località remote selezionate. Per ulteriori informazioni sulle ubicazioni del parco veicoli, consulta. Amazon GameLift Serverssedi di assistenza

In Amazon GameLift Servers Indipendentemente dal tipo di parco veicoli, sei tu a determinare la posizione dell'hardware fisico. Per ulteriori informazioni sulle ubicazioni personalizzate, vederePosizioni per qualsiasi luogo Amazon GameLift Servers.

Istanze On-Demand e istanze Spot

Le istanze Amazon EC2 On-Demand e le istanze Spot offrono lo stesso hardware e le stesse prestazioni, ma differiscono in termini di disponibilità e costi.

Istanze on demand

Puoi acquistare un'istanza on demand quando ne hai bisogno e conservarla per tutto il tempo che desideri. Le istanze on demand hanno un costo fisso, il che significa che paghi per il tempo in cui le utilizzi e non ci sono impegni a lungo termine.

Spot Instances

Le istanze Spot possono offrire un'alternativa conveniente alle istanze on demand utilizzando la capacità di elaborazione inutilizzata. AWS I prezzi delle istanze Spot variano in base alla domanda e all'offerta per ogni tipo di istanza in ogni sede. AWS può interrompere le istanze Spot ogni volta che ha bisogno di recuperare la capacità. Amazon GameLift Servers utilizza le code e il FleetIQ algoritmo per determinare che AWS interromperà un'istanza Spot, mette l'istanza in uno stato di riciclaggio. Quindi, quando non ci sono sessioni di gioco attive sull'istanza, Amazon GameLift Servers tenta di sostituirlo.

Per ulteriori informazioni su come utilizzare le istanze Spot, consultaProgettare una coda per le istanze Spot.

Sistemi operativi

Amazon GameLift Servers le istanze supportano build di server di gioco eseguite su Microsoft Windows o Amazon Linux. Quando carichi una build di gioco su Amazon GameLift Servers, specifica il sistema operativo del gioco. Quando crei una EC2 flotta Amazon per implementare la build del gioco, Amazon GameLift Servers configura automaticamente le istanze con il sistema operativo della build. Per ulteriori informazioni sui sistemi operativi per server di gioco supportati, consultaOttieni strumenti Amazon GameLift Servers di sviluppo.

Quando si utilizza un Amazon GameLift Servers Indipendentemente dal parco macchine, è possibile utilizzare qualsiasi sistema operativo supportato dall'hardware. Amazon GameLift Servers Anywhere, le flotte richiedono che tu distribuisca la tua versione di gioco integrata sull'hardware durante l'utilizzo Amazon GameLift Servers per gestire le tue risorse in un unico posto.

Tipi di istanza

Il tipo di istanza di una EC2 flotta Amazon determina il tipo di hardware utilizzato dalle istanze. Diversi tipi di istanze offrono diverse combinazioni di potenza di calcolo, memoria, archiviazione e funzionalità di rete.

Quando scegli tra i tipi di istanza disponibili per il tuo gioco, considera:

- L'architettura di calcolo del tuo server di gioco: x64 o Arm (AWS Graviton).

> **Note**
>
> Le istanze Graviton Arm richiedono un Amazon GameLift Servers server costruito su sistema operativo Linux. Server SDK 5.1.1 o versione successiva è richiesto per C++ e C#.

Server SDK 5.0 o versione successiva è richiesto per Go. Queste istanze non forniscono out-of-the-box supporto per l'installazione di Mono su Amazon Linux 2023 (AL2023) o Amazon Linux 2 (). AL2

- I requisiti di elaborazione, memoria e archiviazione della build del tuo server di gioco.
- Il numero di processi server che intendi eseguire per istanza.

Utilizzando un tipo di istanza più grande, potresti essere in grado di eseguire più processi server su ciascuna istanza. Ciò può ridurre il numero di istanze necessarie per soddisfare la domanda dei giocatori.

Per ulteriori informazioni:

- Per quanto riguarda i tipi di istanze, consulta Amazon EC2 Instance Types.
- Per informazioni sull'esecuzione di più processi per istanza, consultaGestisci come Amazon GameLift Servers avvia i server di gioco.

Quote del servizio

Per visualizzare le quote di servizio predefinite per Amazon GameLift Serverse le tue quote correnti Account AWS, procedi come segue:

- Per informazioni generali sulle quote di servizio per Amazon GameLift Servers, vedi Amazon GameLift Servers estremi e quote in. Riferimenti generali di AWS
- Per un elenco dei tipi di istanze disponibili per ubicazione per il tuo account, apri la pagina delle quote di servizio del Amazon GameLift Servers console. Questa pagina mostra anche l'utilizzo corrente del tuo account per ogni tipo di istanza in ogni posizione.
- Per un elenco delle quote correnti del tuo account, per esempio i tipi di istanza per regione, esegui il comando AWS Command Line Interface `describe-ec2-instance-limits`(AWS CLI). Questo comando restituisce il numero di istanze attive presenti nella regione predefinita (o in un'altra regione specificata).

Mentre ti prepari a lanciare il gioco, compila un questionario di lancio nel Amazon GameLift Servers console. Il Amazon GameLift Servers il team utilizza il questionario di lancio per determinare le quote e i limiti corretti per il gioco.

Gestisci come Amazon GameLift Servers avvia i server di gioco

Puoi configurare la configurazione di runtime di una EC2 flotta gestita per eseguire più processi del server di gioco per istanza. Questo utilizza le tue risorse di hosting in modo più efficiente.

In che modo un parco istanze gestisce più processi

Amazon GameLift Servers utilizza la configurazione di runtime di una flotta per determinare il tipo e il numero di processi da eseguire su ciascuna istanza. Una configurazione di runtime contiene almeno una configurazione del processo del server che rappresenta un eseguibile del server di gioco. Puoi definire configurazioni aggiuntive dei processi del server per eseguire altri tipi di processi relativi al tuo gioco. Ogni configurazione di processo server contiene le seguenti informazioni:

- Il nome e il percorso del file eseguibile della build di gioco.
- Parametri (facoltativi) da passare al processo all'avvio.
- Il numero di processi da eseguire contemporaneamente.

Quando un'istanza del parco istanze si attiva, avvia il set di processi del server definiti nella configurazione di runtime. Con più processi, Amazon GameLift Servers scagliona l'avvio di ogni processo. I processi server hanno una durata limitata. Quando finiscono, Amazon GameLift Servers avvia nuovi processi per mantenere il numero e il tipo di processi del server definiti nella configurazione di runtime.

Puoi modificare una configurazione di runtime in qualsiasi momento aggiungendo, modificando o rimuovendo le configurazioni del processo del server. Ogni istanza verifica regolarmente la presenza di aggiornamenti alla configurazione di runtime del parco macchine per implementare le modifiche. Ecco come Amazon GameLift Servers adotta le modifiche alla configurazione di runtime:

1. L'istanza invia una richiesta a Amazon GameLift Servers per la versione più recente della configurazione di runtime.
2. L'istanza confronta i processi attivi con la configurazione di runtime più recente e quindi esegue le seguenti operazioni:
   - Se la configurazione di runtime aggiornata rimuove un tipo di processo del server, i processi server attivi di questo tipo continuano a funzionare fino al termine. L'istanza non sostituisce questi processi del server.
   - Se la configurazione di runtime aggiornata riduce il numero di processi simultanei per un tipo di processo server, i processi server in eccesso di questo tipo continuano a funzionare fino al termine. L'istanza non sostituisce questi processi server in eccesso.

- Se la configurazione di runtime aggiornata aggiunge un nuovo tipo di processo del server o aumenta i processi concorrenti per un tipo esistente, l'istanza avvia nuovi processi del server, fino a Amazon GameLift Servers massimo. In questo caso, l'istanza avvia nuovi processi del server al termine dei processi esistenti.

Ottimizza una flotta per più processi

Per utilizzare più processi su una flotta, procedi come segue:

- Crea una build che contenga gli eseguibili del server di gioco che desideri distribuire in una flotta, quindi carica la build su Amazon GameLift Servers. Tutti i server di gioco di una build devono funzionare sulla stessa piattaforma e utilizzare l'SDK del server per Amazon GameLift Servers.
- Creare una configurazione di runtime con una o più configurazioni di processo del server e più processi simultanei.
- Integra i client di gioco con la versione AWS SDK 2016-08-04 o successiva.

Per ottimizzare le prestazioni della flotta, ti consigliamo di fare quanto segue:

- Gestisci gli scenari di arresto dei processi del server in modo che Amazon GameLift Servers può riciclare i processi in modo efficiente. Per esempio:
  - Aggiungi una procedura di spegnimento al codice del server di gioco che richiama l'API del server. `ProcessEnding()`
  - Implementa la funzione di callback `OnProcessTerminate()` nel codice del server di gioco per gestire le richieste di terminazione da Amazon GameLift Servers.
- Assicurati che Amazon GameLift Servers spegne e riavvia i processi del server non integri. Riporta lo stato di salute a Amazon GameLift Servers implementando la funzione di `OnHealthCheck()` callback nel codice del server di gioco. Amazon GameLift Servers chiude automaticamente i processi del server segnalati come non integri per tre report consecutivi. Se non lo `OnHealthCheck()` implementate, allora Amazon GameLift Servers presuppone che un processo server sia integro, a meno che il processo non risponda a una comunicazione.

Scegli il numero di processi per istanza

Quando decidi il numero di processi simultanei da eseguire su un'istanza, tieni presente quanto segue:

- Amazon GameLift Servers limita ogni istanza a un numero massimo di processi simultanei. La somma di tutti i processi simultanei per le configurazioni dei processi del server di una flotta non può superare questa quota.
- Per mantenere livelli di prestazioni accettabili, il tipo di EC2 istanza Amazon potrebbe limitare il numero di processi che possono essere eseguiti contemporaneamente. Prova diverse configurazioni per il tuo gioco per trovare il numero giusto di processi per il tuo tipo di istanza preferito.
- Amazon GameLift Servers non esegue più processi simultanei rispetto al numero totale configurato. Ciò significa che la transizione dalla precedente configurazione di runtime alla nuova configurazione potrebbe avvenire gradualmente.

Aggiornamento di una configurazione del Amazon GameLift Servers parco veicoli

Usa la Amazon GameLift Servers console o la AWS CLI per aggiornare le impostazioni del parco veicoli, modificare le postazioni remote o eliminare un parco veicoli. Per le flotte gestite, non puoi modificare la build o il tipo di istanza del server di gioco di una flotta. Invece, devi sostituire la flotta.

Fast Build Update Tool (solo per lo sviluppo)

Con EC2 le flotte gestite, per implementare un aggiornamento della build di un server di gioco, devi caricare ogni nuova build Amazon GameLift Servers e creare una nuova flotta per essa.
Il Fast Build Update Tool ti consente di aggirare questi passaggi durante lo sviluppo, risparmiando tempo e velocizzando l'iterazione dello sviluppo. Con questo strumento, puoi aggiornare rapidamente i file di build del gioco su tutti i computer di una flotta esistente. Lo strumento offre diverse opzioni: puoi sostituire un'intera build di gioco o modificare 6 file specifici e puoi gestire il riavvio dei processi del server di gioco dopo gli aggiornamenti. Puoi anche usarlo per aggiornare i singoli computer di una flotta.
Per scaricare il Fast Build Update Tool e saperne di più su come usarlo, visita il repository Amazon GameLift Servers Toolkit per The Fast Build Update Tool in Github.

È possibile aggiornare gli attributi mutabili della flotta, le impostazioni delle porte e le configurazioni di runtime utilizzando la Amazon GameLift Servers console o la AWS CLI. Per modificare i limiti di scalabilità, consulta. Scalabilità automatica della capacità della flotta con Amazon GameLift Servers

GameLift Server Amazon console

1. Nella Amazon GameLift Serversconsole, nel pannello di navigazione, scegli Fleets.
2. Scegli la flotta che desideri aggiornare. Una flotta deve essere in `ACTIVE` stato prima di poterla modificare.
3. Nella pagina dei dettagli della flotta, in una delle seguenti sezioni, scegli Modifica.
   - Impostazioni della flotta
     - Modifica gli attributi del parco istanze, ad esempio Name (Nome) e Description (Descrizione).
     - Aggiungi o rimuovi i gruppi di metriche, che Amazon CloudWatch utilizza per tenere traccia delle Amazon GameLift Servers metriche aggregate per più flotte.
     - Aggiorna le impostazioni dei limiti di creazione delle risorse.
     - Attiva o disattiva la protezione delle sessioni di gioco.
   - Configurazione di runtime: è possibile modificare una qualsiasi delle seguenti impostazioni delle configurazioni di runtime e aggiungere o rimuovere configurazioni di runtime.
     - Cambia il percorso di avvio del tuo server di gioco.
     - Aggiungi, rimuovi o modifica i parametri di avvio opzionali.
     - Modifica il numero di processi simultanei eseguiti dai server di gioco.
   - Attivazione della sessione di gioco: modifica il modo in cui desideri che i processi del server vengano eseguiti e ospitino le sessioni di gioco aggiornando il numero massimo di attivazioni simultanee delle sessioni di gioco e il nuovo timeout di attivazione.
   - EC2 impostazioni delle porte: aggiorna gli indirizzi IP e gli intervalli di porte che consentono l'accesso in entrata alla flotta.
4. Scegli Conferma per salvare le modifiche.

AWS CLI

Utilizza i seguenti comandi AWS CLI per aggiornare una flotta:

- update-fleet-attributes
- update-fleet-port-settings
- update-runtime-configuration

Aggiorna le posizioni della flotta

Puoi aggiungere o rimuovere le postazioni remote di una flotta utilizzando la Amazon GameLift Servers console o la AWS CLI. Non puoi modificare la regione di origine di una flotta.

GameLift Server Amazon console

1. Nella Amazon GameLift Serversconsole, nel pannello di navigazione, scegli Fleets.
2. Scegli la flotta che desideri aggiornare. Una flotta deve essere in `ACTIVE` stato prima di poterla modificare.
3. Nella pagina dei dettagli della flotta, scegli la scheda Sedi per visualizzare le sedi della flotta.
4. Per aggiungere nuove postazioni remote, scegli Aggiungi e seleziona le sedi in cui desideri distribuire le istanze. Questo elenco non include le istanze in cui il tipo di istanza del parco istanze non è disponibile.
5. Con le nuove sedi selezionate, scegli Aggiungi. Amazon GameLift Serversaggiunge le nuove sedi all'elenco, con lo stato impostato su`NEW`. Amazon GameLift Serversinizia quindi a fornire un'istanza in ogni posizione aggiunta e a prepararla per ospitare sessioni di gioco.
6. Per rimuovere le postazioni remote esistenti dal parco macchine, utilizza le caselle di controllo per selezionare una o più sedi elencate.
7. Con una o più flotte selezionate, scegli Rimuovi. Le sedi rimosse rimangono nell'elenco, con lo stato impostato `DELETING` su. Amazon GameLift Serversinizia quindi il processo di cessazione dell'attività nella località rimossa. Se ci sono istanze attive che ospitano sessioni di gioco, Amazon GameLift Servers utilizza la procedura di chiusura del server di gioco per terminare correttamente le sessioni di gioco, terminare i server di gioco e chiudere le istanze.

AWS CLI

Utilizza i seguenti comandi AWS CLI per aggiornare le ubicazioni del parco veicoli:

- create-fleet-locations
- delete-fleet-locations

Eliminare un parco veicoli

Puoi eliminare una flotta quando non ne hai più bisogno. L'eliminazione di una flotta rimuove definitivamente tutti i dati associati alle sessioni di gioco e alle sessioni dei giocatori e i dati metrici

raccolti. In alternativa, è possibile mantenere il parco istanze, disattivare il dimensionamento automatico e ridimensionare manualmente il parco a 0 istanze.

> **Note**
>
> Se la flotta dispone di una connessione peering VPC, richiedi prima l'autorizzazione chiamando. CreateVpcPeeringAuthorization Amazon GameLift Serverselimina la connessione peering VPC durante l'eliminazione del parco veicoli.

Puoi utilizzare la Amazon GameLift Servers console o lo strumento AWS CLI per eliminare una flotta.

GameLift Server Amazon console

1. Nella Amazon GameLift Serversconsole, nel pannello di navigazione, scegli Fleets.
2. Scegli la flotta che desideri eliminare. Puoi eliminare solo le flotte presenti nel `ACTIVE` nostro `ERROR` stato.
3. Scegliere Delete (Elimina).
4. Nella finestra di dialogo Elimina flotta, conferma l'eliminazione **`delete`** inserendo.
5. Scegliere Delete (Elimina).

AWS CLI

Utilizza il seguente comando AWS CLI per eliminare una flotta:

- delete-fleet

Debug dei problemi del parco istanze di Amazon GameLift Servers

Questo argomento fornisce indicazioni su come risolvere i problemi relativi alle EC2 flotte Amazon GameLift Servers gestite.

Problemi di creazione del parco istanze

Quando crei una EC2 flotta gestita, il Amazon GameLift Servers servizio avvia un flusso di lavoro che crea la flotta, distribuisce EC2 istanze con la build del server di gioco installata e avvia i processi

del server di gioco su ciascuna istanza. Per una descrizione dettagliata, consulta. Come funziona Amazon GameLift Servers la creazione della flotta Una flotta non può ospitare sessioni di gioco e giocatori finché non raggiunge lo stato Attivo.

Puoi eseguire il debug dei problemi che impediscono alle flotte di diventare attive identificando la fase di creazione della flotta in cui si è verificato il problema ed esaminando gli eventi e i registri di creazione della flotta. Se i registri non offrono informazioni utili, è possibile che il problema sia dovuto a un errore interno del servizio. In questa situazione, prova a creare nuovamente la flotta. Se il problema persiste, prova a caricare nuovamente la build del gioco (per risolvere il possibile danneggiamento del file). Puoi anche contattare il supporto Amazon GameLift Servers o pubblicare una domanda nel forum.

Scaricamento e convalida della build

Durante questa fase, Amazon GameLift Servers ottiene la build del server di gioco caricata, estrae i file ed esegue gli eventuali script di installazione. Se la creazione della flotta fallisce durante queste fasi, consulta gli eventi e i registri della flotta per individuare il problema. Tra le cause possibili sono incluse:

- Amazon GameLift Serversnon riesco a ottenere il file di build compresso (evento). `FLEET_BINARY_DOWNLOAD_FAILED` Verifica che sia possibile accedere alla posizione di archiviazione della build, che tu stia creando una flotta nella Regione AWS stessa della build e che Amazon GameLift Servers disponga delle autorizzazioni corrette per accedervi.
- Amazon GameLift Serversnon riesco a estrarre i file di build (evento`FLEET_CREATION_EXTRACTING_BUILD`).
- Uno script di installazione nei file di build non è stato completato correttamente (evento`FLEET_CREATION_FAILED_INSTALLER`).

Costruire risorse per la flotta

I problemi durante questa fase di solito riguardano l'allocazione e l'impiego delle risorse della flotta. Tra le cause possibili sono incluse:

- Il tipo di istanza richiesto non è disponibile.
- Il tipo di parco veicoli richiesto (Spot o On-Demand) non è disponibile.

Attivazione dei processi del server di gioco

Durante questa fase, Amazon GameLift Servers sta tentando una serie di attività e testando elementi chiave, tra cui la fattibilità del server di gioco, le impostazioni di configurazione del

runtime e la capacità del server di gioco di connettersi al Amazon GameLift Servers servizio tramite Server SDK.

> **Note**
>
> In questa fase, puoi accedere in remoto a un'istanza del parco istanze per esaminare ulteriormente i problemi. Consultare Connessione remota a Amazon GameLift Servers flotte di istanze.

I problemi possibili includono:

- I processi del server non iniziano a funzionare. Ciò suggerisce un problema con le impostazioni di configurazione del runtime del parco macchine (eventi `FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND` o`FLEET_VALIDATION_EXECUTABLE_RUNTIME_FAILURE`. Verifica di aver impostato correttamente il percorso di lancio e i parametri di lancio opzionali.
- I processi del server iniziano a funzionare, ma la flotta non riesce ad attivarsi. Se i processi del server vengono avviati ed eseguiti correttamente, ma il parco macchine non passa allo stato Attivo, è probabile che il processo del server non riesca a comunicare con il Amazon GameLift Servers servizio. Verifica che il server di gioco stia effettuando le chiamate SDK del server corrette (vediInizializza il processo del server):
  - Il processo del server non riesce a inizializzare (evento`SERVER_PROCESS_SDK_INITIALIZATION_TIMEOUT`). Il processo server non sta chiamando `InitSdk()` correttamente.
  - Il processo del server non riesce a notificare Amazon GameLift Servers quando è pronto per ospitare una sessione di gioco (evento`SERVER_PROCESS_PROCESS_READY_TIMEOUT`). Il processo del server è stato inizializzato ma la chiamata non è stata effettuata `ProcessReady()` in tempo.
- Una richiesta di connessione peering VPC non è riuscita. Per i parchi istanze creati con una connessione peering VPC (consulta Per configurare il peering VPC con un nuovo parco istanze), la connessione peering VPC viene effettuata durante la fase Activating (In fase di attivazione). Se una connessione peering VPC non riesce per qualsiasi motivo, il nuovo parco istanze non potrà passare allo stato Active (Attivo). È possibile monitorare l'esito positivo o negativo della richiesta di peering chiamando. describe-vpc-peering-connections Assicurati di verificare che esista un'autorizzazione di peering VPC valida (describe-vpc-peering-authorizations, poiché le autorizzazioni sono valide solo per 24 ore.

Problemi dei processi del server

I processi del server si avviano, ma si interrompono dopo breve tempo o sono poco stabili.

Oltre a problemi con la build del gioco, questo problema si può verificare quanto tenti di eseguire nell'istanza troppi processi del server contemporaneamente. Il numero ottimale di processi contemporanei dipende dal tipo di istanza e dai requisiti di risorse del server di gioco. Prova a ridurre il numero di processi contemporanei, impostati nella configurazione di runtime del parco istanze, per vedere se vi è un miglioramento delle prestazioni. Puoi modificare la configurazione di runtime di una flotta utilizzando la Amazon GameLift Servers console (modifica le impostazioni di allocazione della capacità del parco veicoli) o chiamando il comando. AWS CLI update-runtime-configuration

Problemi di eliminazione del parco istanze

Non è possibile terminare il parco istanze a causa del conteggio istanze massimo.

Il messaggio di errore indica che il parco istanze in fase di eliminazione dispone ancora di istanze attive, situazione non consentita. È prima necessario ridimensionare il parco istanze a zero istanze attive. Questa operazione viene eseguita impostando manualmente il conteggio istanze desiderato del parco istanze su "0" e quindi attendendo che il dimensionamento abbia effetto. Assicurati di disattivare il dimensionamento automatico, operazione che contrasterà le impostazioni manuali.

Le azioni VPC non sono autorizzate.

Questo problema si applica solo alle flotte per le quali sono state create specificamente connessioni peering VPC (vedi. Peering VPC per Amazon GameLift Servers Questo scenario si verifica perché il processo di eliminazione di una flotta include anche l'eliminazione del VPC della flotta e di eventuali connessioni peering VPC. È necessario prima ottenere un'autorizzazione chiamando l'API del servizio for Amazon GameLift Servers CreateVpcPeeringAuthorization() o utilizzando il comando AWS CLI. `create-vpc-peering-authorization` Una volta ottenuta l'autorizzazione, è possibile eliminare il parco istanze.

Amazon GameLift ServersRealtimeproblemi relativi alla flotta

Sessioni di gioco zombie: avviano ed eseguono un gioco, ma non terminano mai.

È possibile riscontrare questo problema, così come uno qualsiasi dei seguenti scenari:

- Gli aggiornamenti script non vengono selezionati dal parco istanze dei server Realtime.
- Il parco istanze raggiunge rapidamente il numero massimo di capacità e non effettua il dimensionamento quando l'attività dei giocatori (ad esempio, nuove richieste di sessione di gioco) diminuisce.

Questa è quasi certamente una conseguenza dell'errore di chiamata di `processEnding` nello script Realtime. Anche se il parco istanze si attiva e le sessioni di gioco vengono avviate, non vi è alcun modo di arrestarle. Di conseguenza, il server Realtime che esegue la sessione di gioco non viene mai liberato così da consentire di avviarne una nuova e le nuove sessioni di gioco possono avviarsi solo quando vengono attivati i nuovi server Realtime. Inoltre, gli aggiornamenti per lo script Realtime non hanno impatto sulle sessioni di gioco già in esecuzione.

Per evitare che si verifichi una situazione simile, gli script devono fornire un meccanismo di attivazione per la chiamata `processEnding`. Come illustrato negli Amazon GameLift ServersRealtimeesempio di script, uno dei modi è programmare un timeout delle sessioni inattive in cui lo script terminerà l'attuale sessione di gioco, se nessun giocatore è connesso per un determinato periodo di tempo.

Tuttavia, se non se non si verifica questo scenario, sono disponibili alcune soluzioni alternative per sbloccare i server Realtime. Il trucco consiste nell'attivare il riavvio dei processi del Realtime server, o delle istanze sottostanti, del parco istanze. In questo caso, chiude automaticamente le sessioni di gioco per te. Amazon GameLift Servers Una volta liberati, i server Realtime possono avviare nuove sessioni di gioco utilizzando la versione più recente dello script Realtime.

Sono disponibili un paio di metodi per ottenere questo risultato, a seconda di quanto è diffuso il problema:

- Diminuire l'intero parco istanze. Questo è il metodo più semplice, ma ha un effetto diffuso. Dimensiona il parco istanze fino a zero, attendi la completa diminuzione del parco istanze, quindi aumentalo nuovamente. Questa operazione cancellerà tutte le sessioni di gioco esistenti e consentirà di avviare lo script Realtime aggiornato più di recente.
- È possibile accedere da remoto all'istanza e riavviare il processo. Si tratta di una buona opzione se il numero di processi da risolvere è ridotto. Se hai già effettuato la registrazione sull'istanza, ad esempio a log di coda o debug, questo potrebbe essere il metodo più rapido. Consultare Connessione remota a Amazon GameLift Servers flotte di istanze.

Se scegli di non includere un modo per chiamare `processEnding` nello script Realtime, esistono un paio di situazioni delicate che potrebbero verificarsi anche quando il parco istanze si attiva e le

sessioni di gioco vengono avviate. In primo luogo, una sessione di gioco in esecuzione non termina. Di conseguenza, il processo del server in cui è in esecuzione tale sessione di gioco non viene mai liberato per consentire di avviare una nuova sessione di gioco. In secondo luogo, il server Realtime non rileva alcun aggiornamento script.

Connessione remota a Amazon GameLift Servers flotte di istanze

Puoi connetterti a qualsiasi istanza attiva Amazon GameLift Servers EC2 flotte gestite. I motivi più comuni per accedere in remoto a un'istanza includono:

- Risolvi i problemi relativi all'integrazione del server di gioco.
- Perfeziona la configurazione di runtime e altre impostazioni specifiche del parco veicoli.
- Ottieni informazioni sulle attività dei server di gioco in tempo reale, ad esempio il tracciamento dei log.
- Esegui strumenti di benchmarking utilizzando il traffico effettivo dei giocatori.
- Esamina problemi specifici relativi a una sessione di gioco o a un processo del server.

Quando ti connetti a un'istanza, considera questi potenziali problemi:

- Puoi connetterti a qualsiasi istanza in un parco istanze attivo. In genere, non è possibile connettersi a flotte non attive, ad esempio flotte in fase di attivazione o in stato di errore. (Queste flotte potrebbero avere una disponibilità limitata per un breve periodo di tempo.) Per assistenza con i problemi di attivazione della flotta, consultaDebug dei problemi del parco istanze di Amazon GameLift Servers.
- La connessione a un'istanza attiva non influisce sull'attività di hosting dell'istanza. L'istanza continua ad avviare e arrestare i processi del server in base alla configurazione di runtime. Attiva ed esegue sessioni di gioco. L'istanza potrebbe chiudersi in risposta a un evento di ridimensionamento o a un altro evento.
- Qualsiasi modifica apportata ai file o alle impostazioni sull'istanza potrebbe influire sulle sessioni di gioco attive dell'istanza e sui giocatori connessi.

Le seguenti istruzioni descrivono come connettersi in remoto a un'istanza utilizzando l'interfaccia a riga di AWS comando (CLI). Puoi anche effettuare chiamate programmatiche utilizzando l' AWS SDK, come documentato nel riferimento all'API di servizio per Amazon GameLift Servers.

Raccogli dati delle istanze

Per connettersi a un Amazon GameLift Servers un'istanza di EC2 flotta gestita, sono necessarie le seguenti informazioni:

- L'ID dell'istanza a cui desideri connetterti. È possibile utilizzare l'ID dell'istanza o l'ARN.
- l'SDK del server per Amazon GameLift Servers versione utilizzata sull'istanza. L'SDK del server è integrato con la build del gioco in esecuzione sull'istanza.

Le seguenti istruzioni descrivono come completare queste attività utilizzando la AWS CLI. Devi conoscere l'ID della flotta per l'istanza a cui desideri connetterti.

1. Ottieni il nome del computer. Ottieni un elenco di tutti i computer attivi della flotta. Calcola l'elenco delle chiamate con un Fleet ID o un ARN. Per una flotta con una sola sede, specifica solo l'identificatore della flotta. Per una flotta con più sedi, specifica l'identificatore della flotta e una posizione. Con le EC2 flotte gestite, `list-compute` restituisce un elenco di istanze del parco istanze e la proprietà `ComputeName` è l'ID dell'istanza. Trova il computer a cui desideri accedere.

   Richiesta

   ```
   aws gamelift list-compute \
     --fleet-id  "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa" \
     --location ""sa-east-1"
   ```

   Risposta

   ```
   {
     "ComputeList": [
       {
         "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
         "FleetArn": "arn:aws:gamelift:us-west-2::fleet/
   fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
         "ComputeName": "i-0abc12d3e45fa6b78",
         "IpAddress": "00.00.000.00",
         "DnsName":
    "b08444ki909kvqu6zpw3is24x5pyz4b6m05i3jbxvpk9craztu0lqrbbrbnbkks.uwp57060n1k6dnlnw49b78hg1
   west-2.amazongamelift.com",
         "ComputeStatus": "Active",
         "Location": "sa-east-1",
   ```

```
      "CreationTime": "2023-07-09T22:51:45.931000-07:00",
      "OperatingSystem": "AMAZON_LINUX_2023",
      "Type": "c4.large"
    }
  ]
}
```

2. Trova la versione SDK del server. Per queste informazioni devi cercare la build che viene implementata nella flotta. La versione Server SDK è una proprietà di compilazione.

   a. Chiama describe-fleet-attributescon un ID flotta o un ARN per ottenere l'ID di build e l'ARN della flotta.

   b. Chiama describe-build con l'ID di build o l'ARN per ottenere la versione SDK del server della build.

      Per esempio:

      Richiesta

```
aws gamelift describe-fleet-attributes /
  --fleet-ids  "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa"
```

Risposta

```
{
  "FleetAttributes": [
    {
      "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
      "ComputeType": "EC2",
      "BuildId": "build-3333cccc-44dd-55ee-66ff-00001111aa22",
      . . .
    }
  ]
}
```

Richiesta

```
aws gamelift describe-build /
  --build-id "build-3333cccc-44dd-55ee-66ff-00001111aa22"
```

Risposta

```
"Build": {
  "BuildId": "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
  "Name": "My_Game_Server_Build_One",
 "OperatingSystem": "AMAZON_LINUX_2023",

  "ServerSdkVersion": "5.1.1",
  . . .
}
```

Connect a un'istanza (server SDK 5)

Se l'istanza a cui vuoi connetterti esegue una build di gioco con server SDK versione 5.x, connettiti all'istanza utilizzando Amazon EC2 Systems Manager (SSM). Puoi accedere alle istanze remote in esecuzione su Windows o Linux.

> **Prima di iniziare:**
>
> Completa i passaggi di configurazione SSM e installa il plug-in SSM sul tuo computer locale. Per ulteriori informazioni, consulta Configurazione di SSM e Installazione del plug-in Session Manager per la AWS CLI nella Amazon Systems EC2 Manager User Guide.

1. Richiedere le credenziali di accesso all'istanza. Chiama get-compute-accesscon l'ID della flotta e il nome di calcolo dell'istanza a cui desideri connetterti. Amazon GameLift Servers restituisce un set di credenziali temporanee per accedere all'istanza. Per esempio:

   Richiesta

   ```
   aws gamelift get-compute-access \
   --compute-name i-11111111a222b333c \
   --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa
   --region us-west-2
   ```

   Risposta

   ```
   {
   ```

```
  "ComputeName": " i-11111111a222b333c ",
  "Credentials": {
    "AccessKeyId": " ASIAIOSFODNN7EXAMPLE ",
    "SecretAccessKey": " wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY ",
    "SessionToken": " AQoDYXdzEJr...<remainder of session token>"
  },
  "FleetArn": " arn:aws:gamelift:us-west-2::fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa ",
  "FleetId": " fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa "
}
```

2. Esporta le credenziali di accesso (opzionale). È possibile esportare le credenziali in variabili di ambiente e utilizzarle per configurare la AWS CLI per l'utente predefinito. Per maggiori dettagli, consulta Variabili di ambiente per configurare la AWS CLI nella Guida per l' AWS Command Line Interface utente.

```
export AWS_ACCESS_KEY_ID=ASIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of session token>
```

3. Connect all'istanza fleet. Avvia una sessione SSM con l'istanza a cui desideri connetterti. Includi la AWS regione o la posizione dell'istanza. Per ulteriori informazioni, tra cui come configurare SSM e il plug-in SSM, consulta Avvio di una sessione (AWS CLI) nella Amazon EC2 Systems Manager User Guide.

   La richiesta di avvio della sessione utilizzerà automaticamente le credenziali acquisite nella fase 1.

```
aws ssm start-session \
--target i-11111111a222b333c \
--region us-west-2 \
```

> **Note**
>
> Se viene visualizzato un errore di accesso negato, è possibile che una variabile di `AWS_PROFILE` ambiente sia impostata AWS su un profilo, il che fa sì che AWS CLI utilizzi le credenziali errate per l'accesso remoto. Per risolvere, disattiva temporaneamente la variabile di ambiente`AWS_PROFILE`. In alternativa, è possibile

creare un AWS profilo personalizzato per le credenziali di accesso remoto e aggiungere il parametro della riga di `--profile` comando alla richiesta`start-session`.

Connect a un'istanza (server SDK 4.x o precedente)

Se l'istanza a cui vuoi connetterti esegue una build di gioco con server SDK versione 4 o precedente, usa le seguenti istruzioni. Puoi connetterti a istanze che eseguono Windows o Linux. Connect a un'istanza Windows utilizzando un client RDP (Remote Desktop Protocol). Connect a un'istanza Linux utilizzando un client SSH.

1. Richiedere le credenziali di accesso all'istanza. Se disponi di un ID di istanza, usa il comando get-instance-accessper richiedere le credenziali di accesso. In caso di successo, Amazon GameLift Servers restituisce il sistema operativo dell'istanza, l'indirizzo IP e un set di credenziali (nome utente e chiave segreta). Il formato delle credenziali dipende dal sistema operativo dell'istanza. Utilizzare le istruzioni seguenti per recuperare le credenziali per RDP o SSH.

   - Per le istanze Windows: per connettersi a un'istanza Windows, RDP richiede un nome utente e una password. La richiesta `get-instance-access` restituisce tali valori come stringhe semplici, pertanto è possibile utilizzare i valori restituiti così come sono. Credenziali di esempio:

     ```
     "Credentials": {
         "Secret": "aA1bBB2cCCd3EEE",
         "UserName": "gl-user-remote"
     }
     ```

   - Per le istanze Linux: per connettersi a un'istanza Linux, SSH richiede un nome utente e una chiave privata. Amazon GameLift Servers emette chiavi private RSA e le restituisce come una singola stringa, con il carattere newline (\n) che indica le interruzioni di riga. Per rendere utilizzabile la chiave privata, procedi nel seguente modo: (1) converti la stringa in un `.pem` file e (2) imposta le autorizzazioni per il nuovo file. Credenziali di esempio restituite:

     ```
     "Credentials": {
         "Secret": "-----BEGIN RSA PRIVATE KEY-----
     nEXAMPLEKEYKCAQEAy7WZhaDsrA1W3mRlQtvhwyORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/
     \nvBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7iO5dSrvC7dQkW2duV5QuUdE0QW
     \nZ/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449ilx9X1F
     \nG50TCFeOzfl8dqqCP6GzbPaIjiU19xX/azOR9V+tpUOzEL+wmXnZt3/nHPQ5xvD2OJH67km6SuPW
     ```

```
\noPzev/D8V+x4+bHthfSjR9Y7DvQFjfBVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrqu
\n/uler7vgIn5m7lN5LKw4hJLAIW6tUT/fzvtcHK0SkbQCQXuriHmQ2MQyJX/0kn2NfjLV/
ufGxbL1\nmb5qwMGUnEpJaZD6QSSs3kICLwWUYUiGfc0uiSbmJoap/
GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2\nbahyWyJNfjLe4M86yd2YK3V2CmK+X/
BOsShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/
tJWSD9\n81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA3OzdXzMQexXVJ1TLZVEH0E7bhlY9d8O1ozR
\noQs/FiZNAx2iijCWyv0lpjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfql
+lIp1\nYkriL0DbLXlvRAH+yHPRit2hHOjtUNZh4Axv+cpg09qbUI3+43eEy24B7G/Uh
+GTfbjsXsOxQx/x\np9otyVwc7hsQ5TA5PZb
+mvkJ5OBEKzet9XcKwONBYELGhnEPe7cCgYEA06Vgov6YHleHui9kHuws
\nayav0elc5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URrpzWVOEihvm+xTtmaZlSp//lkq75XDwnU
\nWA8gkn6O3QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
\ngYBjbO+OZk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9PfN295yQ+BxMBXiIycWVQiw0bH
\noMo7yykABY7Ozd5wQewBQ4AdSlWSX4nGDtsiFxWiI5sKuAAeOCbTosy1s8w8fxoJ5Tz1sdoxNeGs
\nArq6Wv/G16zQuAE9zK9vvwKBgF+09VI/1wJBirsDGz9whVWfFPrTkJNvJZzYt69qezxlsjgFKshy
\nWBhd4xHZtmCqpBPlAymEjr/TOlbxyARmXMnIOWIAnNXMGB4KGSyl1mzSVAoQ+fqR+cJ3d0dyPl1j
\njjb0Ed/NY8frlNDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0iOegLDa
\nNWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvtW+nY2jVhv7UGd8MjwUTNGItdb6nsYqM2asrnF3qS
\nVRkAKKKYeGjkpUfVTrW0YFjXkfcrR/V+QFL5OndHAKJXjW7a4ejJLncTzmZSpYzwApc=\n-----END
 RSA PRIVATE KEY-----",
    "UserName": "gl-user-remote"
}
```

Quando si utilizza la AWS CLI, è possibile generare automaticamente un `.pem` file includendo i parametri --query e --output nella richiesta. `get-instance-access`

Per impostare le autorizzazioni sul file `.pem`, eseguire il comando seguente:

```
$ chmod 400 MyPrivateKey.pem
```

2. Aprire una porta per la connessione remota. Puoi accedere alle istanze in Amazon GameLift Servers flotte attraverso qualsiasi porta autorizzata nella configurazione della flotta. È possibile visualizzare le impostazioni di porta del parco istanze tramite il comando [describe-fleet-port-settings](#).

   Come best practice, si consiglia di aprire le porte per l'accesso remoto solo quando necessario e di chiuderle al termine dell'operazione. Non puoi aggiornare le impostazioni delle porte dopo aver creato una flotta ma prima che sia attiva. Se rimani bloccato, ricrea la flotta con le impostazioni del porto aperte.

Utilizzare il comando `update-fleet-port-settings` per aggiungere un'impostazione di porta per la connessione remota (ad esempio 22 per SSH o 3389 per RDP). Per il valore di intervallo IP, specificare gli indirizzi IP dei dispositivi che si intende utilizzare per la connessione (convertiti in formato CIDR). Esempio:

```
$ AWS gamelift update-fleet-port-settings
    --fleet-id  "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa"
    --inbound-permission-authorizations
 "FromPort=22,ToPort=22,IpRange=54.186.139.221/32,Protocol=TCP"
```

L'esempio seguente apre la porta 3389 su una flotta di Windows

```
$ AWS gamelift update-fleet-port-settings
--fleet-id  "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa"
    --inbound-permission-authorizations
 "FromPort=3389,ToPort=3389,IpRange=54.186.139.221/32,Protocol=TCP"
```

3. Aprire un client di connessione remota. Utilizzare Desktop remoto per le istanze Windows o SSH per le istanze Linux. Connettersi all'istanza utilizzando l'indirizzo IP, l'impostazione di porta e le credenziali di accesso.

   Esempio SSH:

   ```
   ssh -i MyPrivateKey.pem gl-user-remote@192.0.2.0
   ```

Visualizza i file su istanze remote

Quando ti connetti in remoto a un'istanza, disponi dell'accesso utente e amministratore completo. Ciò vuol dire che puoi inoltre causare errori e guasti nell'hosting di gioco. Se l'istanza ospita giochi con giocatori attivi, corri il rischio di interrompere le sessioni di gioco e far cadere i giocatori, oppure di interrompere i processi di chiusura del gioco e causare errori nei dati e nei registri di gioco salvati.

Cerca queste risorse su un'istanza di hosting:

- File della build di gioco. Questi file sono la build del gioco su cui hai caricato Amazon GameLift Servers. Includono uno o più file eseguibili, risorse e dipendenze del server di gioco. I file di build del gioco si trovano in una directory principale chiamata: `game`
  - Su Windows: `c:\game`

  - In Linux: `/local/game`
- File di log del gioco. Trova i file di registro generati dal tuo server di gioco nella directory `game` principale in qualsiasi percorso di directory che hai designato.
- Amazon GameLift Servers risorse di hosting. La directory principale `Whitewater` contiene i file utilizzati da Amazon GameLift Servers servizio per gestire l'attività di hosting dei giochi. Non modificare questi file per nessun motivo.
- Configurazione di runtime. Non accedete alla configurazione di runtime per le singole istanze. Per apportare modifiche a una proprietà di configurazione di runtime, aggiorna la configurazione di runtime della flotta (vedi il funzionamento dell' AWS SDK UpdateRuntimeConfigurationo il AWS CLI update-runtime-configuration).
- Dati sulla flotta. Un file JSON contiene informazioni sulla flotta a cui appartiene l'istanza, utilizzabili dai processi del server in esecuzione sull'istanza. Il file JSON si trova nella seguente posizione:
  - Su Windows: `C:\GameMetadata\gamelift-metadata.json`
  - In Linux: `/local/gamemetadata/gamelift-metadata.json`
- Certificati TLS. Se l'istanza fa parte di una flotta in cui è abilitata la generazione di certificati TLS, cerca i file di certificato, tra cui il certificato, la catena di certificati, la chiave privata e il certificato principale nella seguente posizione:
  - Su Windows: `c:\\GameMetadata\Certificates`
  - In Linux: `/local/gamemetadata/certificates/`

Peering VPC per Amazon GameLift Servers

Questo argomento fornisce indicazioni su come configurare una connessione peering VPC tra Amazon GameLift Servers-server di gioco ospitati e altri server non ospitatiAmazon GameLift Servers risorse. Utilizza le connessioni peering Amazon Virtual Private Cloud (VPC) per consentire ai server di gioco di comunicare direttamente e privatamente con le altre AWS risorse, come un servizio Web o un repository. Puoi stabilire il peering VPC con qualsiasi risorsa in esecuzione AWS e gestita da un AWS account a cui hai accesso.

> **Note**
>
> Peering VPC è una caratteristica avanzata. Per maggiori informazioni sulle opzioni preferite per consentire ai server di gioco di comunicare direttamente e privatamente con le altre AWS risorse, consulta. Comunica con altre AWS risorse delle tue flotte

Se conosci già il peering di Amazon VPCs e VPC, tieni presente che configurare il peering con Amazon GameLift Servers i server di gioco sono leggermente diversi. Non hai accesso al VPC che contiene i tuoi server di gioco: è controllato dal Amazon GameLift Servers servizio, quindi non puoi richiedere direttamente il peering VPC per questo. Invece, devi prima pre-autorizzare il VPC con il tuo codice non-Amazon GameLift Servers risorse per accettare una richiesta di peering proveniente da Amazon GameLift Servers servizio. Quindi si attiva Amazon GameLift Servers per richiedere il peering VPC che hai appena autorizzato. Amazon GameLift Servers gestisce le attività di creazione della connessione peering, impostazione delle tabelle di routing e configurazione della connessione.

Configurazione del peering VPC per un parco istanze esistente

1. Ottieni gli ID e le credenziali dell' AWS account.

   Sono necessari un ID e le credenziali di accesso per i seguenti account. AWS Puoi trovare l' AWS account IDs accedendo AWS Management Consolee visualizzando le impostazioni dell'account. Per ottenere le credenziali, andare alla console IAM.

   - AWS account che utilizzi per gestire il tuo Amazon GameLift Servers server di gioco.
   - AWS account che usi per gestire i tuoi account non-Amazon GameLift Servers risorse.

   Se utilizzi lo stesso account per Amazon GameLift Servers e non-Amazon GameLift Servers risorse, sono necessari ID e credenziali solo per quell'account.

2. Ottenimento di identificatori per ogni VPC.

   Ottieni le seguenti informazioni per consentire il peering dei due VPCs :

   - VPC per il tuo Amazon GameLift Servers server di gioco: questo è il tuo Amazon GameLift Servers ID della flotta. I tuoi server di gioco sono distribuiti in Amazon GameLift Servers su una flotta di EC2 istanze. Una flotta viene inserita automaticamente nel proprio VPC, gestito dal Amazon GameLift Servers servizio. Non hai accesso diretto al VPC, quindi è identificato dall'ID della flotta.
   - VPC per chi non...Amazon GameLift Servers AWS risorse: puoi stabilire un peering VPC con qualsiasi risorsa in esecuzione AWS e gestita da un AWS account a cui hai accesso. Se non hai ancora creato un VPC per queste risorse, consulta la Guida introduttiva ad Amazon VPC. Dopo aver creato un VPC, puoi trovare l'ID VPC accedendo a AWS Management ConsoleAmazon VPC e visualizzando il tuo. VPCs

> **Note**
>
> Quando si configura un peering, entrambi VPCs devono esistere nella stessa regione. Il VPC per te Amazon GameLift Servers i server di gioco della flotta si trovano nella stessa regione della flotta.

3. Autorizza un peering VPC.

   In questo passaggio, stai pre-autorizzando una richiesta futura da Amazon GameLift Servers per collegare il VPC ai tuoi server di gioco con il tuo VPC per non-Amazon GameLift Servers risorse. Questa operazione aggiorna il gruppo di sicurezza per il VPC.

   Per autorizzare il peering VPC, chiama il service CreateVpcPeeringAuthorizationAPI () o usa il comando CLI AWS . `create-vpc-peering-authorization` Effettua questa chiamata utilizzando l'account che gestisce la tua rete non-Amazon GameLift Servers risorse. Identificare le informazioni che seguono:

   - ID VPC peer: è per il VPC con il tuo non-Amazon GameLift Servers risorse.
   - Amazon GameLift Servers AWS ID account: questo è l'account che usi per gestire il tuo Amazon GameLift Servers flotta.

   Una volta autorizzato un peering VPC, l'autorizzazione rimane valida per 24 ore a meno che non venga revocata. È possibile gestire le autorizzazioni peer VPC utilizzando le operazioni descritte di seguito:

   - DescribeVpcPeeringAuthorizations() (AWS CLI`describe-vpc-peering-authorizations`).
   - DeleteVpcPeeringAuthorization() (AWS CLI`delete-vpc-peering-authorization`).
4. Richiedi una connessione peering.

   Con un'autorizzazione valida, puoi richiederlo Amazon GameLift Servers stabilire una connessione peering.

   Per richiedere un peering VPC, chiama il service API CreateVpcPeeringConnection() o usa il comando AWS CLI. `create-vpc-peering-connection` Effettua questa chiamata

utilizzando l'account che gestisce il Amazon GameLift Servers server di gioco. Utilizza le seguenti informazioni per identificare i due VPCs che desideri utilizzare come peer:

- ID VPC peer AWS e ID account: questo è il VPC per utenti non appartenenti aAmazon GameLift Servers risorse e l'account che usi per gestirle. L'ID VPC deve corrispondere all'ID di un'autorizzazione peering valida.
- Fleet ID: identifica il VPC per il tuo Amazon GameLift Servers server di gioco.

5. Monitora lo stato della connessione peering.

   Richiedere una connessione peer VPC è un'operazione asincrona. Per monitorare lo stato di una richiesta di peer e gestire i casi di successo o di errore, utilizzare una delle seguenti opzioni:

   - Effettuare continuamente il polling con `DescribeVpcPeeringConnections()`. Questa operazione recupera il record di connessione peer VPC, tra cui lo stato della richiesta. Se una connessione peer viene creata correttamente, il record di connessione contiene anche un blocco CIDR di indirizzi IP privati assegnato al VPC.
   - Gestisci gli eventi della flotta associati alle connessioni di peering VPC con DescribeFleetEvents(), inclusi gli eventi di successo e di fallimento.

Una volta stabilita la connessione peering, puoi gestirla utilizzando le seguenti operazioni:

- DescribeVpcPeeringConnections() (AWS CLI`describe-vpc-peering-connections`).
- DeleteVpcPeeringConnection() (AWS CLI`delete-vpc-peering-connection`).

Per configurare il peering VPC con un nuovo parco istanze

Puoi crearne uno nuovo Amazon GameLift Servers flotta e richiedi contemporaneamente una connessione peering VPC.

1. Ottieni gli ID e le credenziali dell' AWS account.

   Sono necessari un ID e le credenziali di accesso per i due account seguenti. AWS Puoi trovare l' AWS account IDs accedendo AWS Management Consolee visualizzando le impostazioni dell'account. Per ottenere le credenziali, andare alla console IAM.

   - AWS account che utilizzi per gestire il tuo Amazon GameLift Servers server di gioco.
   - AWS account che usi per gestire i tuoi account non-Amazon GameLift Servers risorse.

Se utilizzi lo stesso account per Amazon GameLift Servers e non-Amazon GameLift Servers risorse, sono necessari ID e credenziali solo per quell'account.

2. Ottieni l'ID VPC per i tuoi utenti non-Amazon GameLift ServersAWS risorse.

   Se non hai ancora creato un VPC per queste risorse, fallo ora (vedi Guida introduttiva ad Amazon VPC). Assicurati di creare il nuovo VPC nella stessa regione in cui prevedi di creare il nuovo parco istanze. Se non seiAmazon GameLift Servers le risorse sono gestite con un AWS account o un gruppo di utenti/utenti diverso da quello con cui utilizzi Amazon GameLift Servers, dovrai utilizzare queste credenziali dell'account quando richiedi l'autorizzazione nel passaggio successivo.

   Dopo aver creato un VPC, puoi individuare l'ID VPC nella console Amazon VPC visualizzando il tuo. VPCs

3. Autorizza un peering VPC senza-Amazon GameLift Servers risorse.

   Quando Amazon GameLift Servers crea la nuova flotta e un VPC corrispondente, inoltre invia una richiesta di peer con il VPC per i tuoi utenti non-Amazon GameLift Servers risorse. Devi preautorizzare la richiesta. Questa fase aggiorna il gruppo di sicurezza per il VPC.

   Utilizzo delle credenziali dell'account che gestiscono i dati non-Amazon GameLift Servers risorse, chiama l'API del servizio CreateVpcPeeringAuthorization() o usa il comando AWS CLI. `create-vpc-peering-authorization` Identificare le informazioni che seguono:

   - Peer VPC ID: ID del VPC con il tuo non-Amazon GameLift Servers risorse.
   - Amazon GameLift Servers AWS ID account: ID dell'account che utilizzi per gestire il Amazon GameLift Servers flotta.

   Una volta autorizzato un peering VPC, l'autorizzazione rimane valida per 24 ore a meno che non venga revocata. È possibile gestire le autorizzazioni peer VPC utilizzando le operazioni descritte di seguito:

   - DescribeVpcPeeringAuthorizations() (AWS CLI`describe-vpc-peering-authorizations`).
   - DeleteVpcPeeringAuthorization() (AWS CLI`delete-vpc-peering-authorization`).

4. Segui le istruzioni per creare una nuova flotta utilizzando la AWS CLI. Includere i seguenti parametri aggiuntivi:

- peer-vpc-aws-account-id — ID dell'account che usi per gestire il VPC con il tuo account diverso da-Amazon GameLift Servers risorse.
- peer-vpc-id— ID del VPC con il tuo codice non-Amazon GameLift Servers conto.

Una chiamata effettuata correttamente a create-fleet con i parametri di peering VPC genera un nuovo parco istanze e una nuova richiesta di peering VPC. Lo stato del parco istanze è impostato su New (Nuovo) e il processo di attivazione del parco istanze viene avviato. Lo stato della richiesta di connessione peering è impostato su initiating-request. È possibile monitorare l'esito positivo o negativo della richiesta di peering describe-vpc-peering-connectionschiamando.

Quando richiedi un nuovo parco istanze e una connessione peering VPC, entrambe le operazioni possono riuscire correttamente o generare un errore. Se un parco istanze genera un errore durante il processo di creazione, la connessione VPC in peering non verrà stabilita. Analogamente, se una connessione peering VPC non riesce per qualsiasi motivo, il nuovo parco istanze non potrà passare dallo stato Activating (In attivazione) allo stato Active (Attivo).

> **Note**
>
> La nuova connessione peering VPC non viene completata finché il parco istanze non è pronto per l'attivazione. Ciò significa che la connessione non è disponibile e non può essere utilizzata durante il processo di installazione della build del server di gioco.

Nell'esempio seguente viene creato un nuovo parco istanze e una connessione peering tra un VPC prestabilito e il VPC per il nuovo parco istanze. Il VPC prestabilito viene identificato in modo univoco dalla combinazione di dati non-Amazon GameLift Servers AWS l'ID dell'account e l'ID VPC.

```
$ AWS gamelift create-fleet
    --name "My_Fleet_1"
    --description "The sample test fleet"
    --ec2-instance-type "c5.large"
    --fleet-type "ON_DEMAND"
    --build-id "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff"
    --runtime-configuration "GameSessionActivationTimeoutSeconds=300,
                             MaxConcurrentGameSessionActivations=2,
                             ServerProcesses=[{LaunchPath=C:\game\Bin64.dedicated
\MultiplayerSampleProjectLauncher_Server.exe,
                                                Parameters=+sv_port 33435 +start_lobby,
```

```
                                                ConcurrentExecutions=10}]"
    --new-game-session-protection-policy "FullProtection"
    --resource-creation-limit-policy "NewGameSessionsPerCreator=3,
                                      PolicyPeriodInMinutes=15"
    --ec2-inbound-permissions
 "FromPort=33435,ToPort=33435,IpRange=0.0.0.0/0,Protocol=UDP"

 "FromPort=33235,ToPort=33235,IpRange=0.0.0.0/0,Protocol=UDP"
    --metric-groups  "EMEAfleets"
    --peer-vpc-aws-account-id "111122223333"
    --peer-vpc-id "vpc-a11a11a"
```

Versione copiabile:

```
AWS gamelift create-fleet --name "My_Fleet_1" --description "The
 sample test fleet" --fleet-type "ON_DEMAND" --metric-groups
 "EMEAfleets" --build-id "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff"
 --ec2-instance-type "c5.large" --runtime-configuration
 "GameSessionActivationTimeoutSeconds=300,MaxConcurrentGameSessionActivations=2,ServerProcesses
\game\Bin64.dedicated\MultiplayerSampleProjectLauncher_Server.exe,Parameters=
+sv_port 33435 +start_lobby,ConcurrentExecutions=10}]" --new-game-session-
protection-policy "FullProtection" --resource-creation-limit-policy
 "NewGameSessionsPerCreator=3,PolicyPeriodInMinutes=15" --ec2-inbound-
permissions "FromPort=33435,ToPort=33435,IpRange=0.0.0.0/0,Protocol=UDP"
 "FromPort=33235,ToPort=33235,IpRange=0.0.0.0/0,Protocol=UDP" --peer-vpc-aws-account-id
 "111122223333" --peer-vpc-id "vpc-a11a11a"
```

Risoluzione dei problemi di peering VPC

Se hai problemi a stabilire una connessione peering VPC per il tuo Amazon GameLift Servers server di gioco, considera queste cause principali comuni:

- Non è stata trovata alcuna autorizzazione per la connessione richiesta:
  - Verifica lo stato di un'autorizzazione VPC per i non-Amazon GameLift Servers VPC. Potrebbe non esistere o essere scaduto.
  - Controlla le regioni delle due VPCs che stai cercando di scrutare. Se non si trovano nella stessa regione, non possono essere sottoposti al peering.
- I blocchi CIDR (vedi Configurazioni di connessione peering VPC non valide) dei due blocchi si sovrappongono. VPCs I blocchi IPv4 CIDR assegnati al peered non possono sovrapporsi. VPCs Il blocco CIDR del VPC per il tuo Amazon GameLift Servers la flotta viene assegnata

automaticamente e non può essere modificata, quindi dovrai modificare la forma del blocco CIDR del VPC per i tuoi utenti non-Amazon GameLift Servers risorse. Per risolvere il problema:

- Cerca questo blocco CIDR per il tuo Amazon GameLift Servers flotta `DescribeVpcPeeringConnections()` chiamando.
- Vai alla console Amazon VPC e trova il VPC per il tuo account non-Amazon GameLift Servers risorse e modifica il blocco CIDR in modo che non si sovrappongano.

- Il nuovo parco istanze non è attivato (quando si richiede il peering VPC con un nuovo parco istanze). Se il nuovo parco istanze non è riuscito a passare allo stato Attivo non vi è alcun VPC da sottoporre al peering, pertanto la connessione peering non può avere esito positivo.

Amazon GameLift Serversflotte di container gestite

Amazon GameLift Serversle flotte di container gestite offrono una piattaforma basata su cloud per ospitare il software del server di gioco containerizzato. Con una flotta di container, ottieni la flessibilità, la sicurezza e l'affidabilità delle Cloud AWS risorse, ulteriormente ottimizzate per l'hosting di giochi multiplayer. Il Amazon GameLift Servers servizio fornisce solidi strumenti di gestione degli host.

Accelera l'onboarding con questi strumenti per container gestiti:

- Lo starter kit per container semplifica l'integrazione e la configurazione della flotta. Aggiunge funzionalità essenziali per la gestione delle sessioni di gioco al server di gioco e utilizza modelli preconfigurati per creare una flotta di container e una pipeline di distribuzione automatizzata per il server di gioco. Dopo l'implementazione, utilizza la Amazon GameLift Servers console e gli strumenti API per monitorare le prestazioni della flotta, gestire le sessioni di gioco e analizzare le metriche.
- Per gli sviluppatori di Unreal Engine o Unity, usa i Amazon GameLift Serversplugin e il server del motore di gioco SDKs per integrare il tuo server di gioco e creare una flotta di container dall'interno dell'ambiente di sviluppo del tuo motore di gioco. I flussi di lavoro guidati del plug-in ti aiutano a creare una soluzione semplice e veloce con l'hosting basato su cloud utilizzando contenitori gestiti. Puoi basarti su queste basi per creare una soluzione di hosting personalizzata per il tuo gioco.

Una flotta di container gestita è costituita da un set di computer virtuali che Amazon GameLift Servers possiedono e operano per conto dell'utente e in base alle scelte di configurazione dell'utente. I computer sono istanze Amazon Elastic Compute Cloud EC2 (Amazon) con Amazon GameLift Servers sistema operativo Linux. Le istanze si trovano fisicamente nelle nostre Regioni AWS Local Zones. Quando crei una flotta di container, scegli un tipo di EC2 istanza per i tuoi computer in base a potenza di calcolo, memoria, archiviazione, funzionalità di rete e altri fattori.

Per una flotta di container gestita, memorizzi le immagini dei container basate su Linux in un repository Amazon Elastic Container Registry (Amazon ECR) e crei una definizione di gruppo di contenitori per descrivere l'architettura dei container. Quando crei una flotta, il Amazon GameLift Servers servizio utilizza la definizione del gruppo di contenitori per distribuire le immagini dei container nelle istanze del parco veicoli. Quando i container avviano i processi del server di gioco, ogni processo stabilisce una connessione al Amazon GameLift Servers servizio e segnala la disponibilità a ospitare una sessione di gioco.

Oltre alla distribuzione della flotta, Amazon GameLift Servers gestisce le seguenti attività di gestione dell'host in modo da non dover:

- Monitora lo stato di tutti i container della flotta e sostituisce quelli obsoleti o insalubri.
- Gestisce l'autenticazione per la comunicazione tra i processi del server e il servizio. Amazon GameLift Servers
- Offre strumenti di auto-scaling che regolano la capacità della flotta in modo dinamico per soddisfare la domanda dei giocatori.
- Riporta le metriche delle prestazioni per le EC2 istanze, i container e i processi server della flotta.

Consulta questi argomenti su come configurare e gestire flotte di container gestite:

- Roadmap di sviluppo per l'hosting con contenitori Amazon GameLift Servers gestiti
- Crea un Amazon GameLift Servers flotta di container gestita
- Personalizza una flotta di Amazon GameLift Servers container
- Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers
- Aggiorna un Amazon GameLift Servers flotta di container gestita

Crea un Amazon GameLift Servers flotta di container gestita

Crea un Amazon GameLift Servers flotta di container gestita per distribuire e ospitare il tuo server di gioco containerizzato nel cloud. AWS Quando crei una flotta di container, puoi specificare le definizioni dei gruppi di container con le impostazioni di configurazione, tra cui una o più immagini di container, inclusa un'immagine con il tuo server di gioco. Puoi anche scegliere di creare una flotta di container vuota e aggiungere o aggiornare le definizioni dei gruppi di container della flotta in un secondo momento.

Puoi usare il Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI) per creare una flotta di container.

Dopo aver creato una nuova flotta di container, lo stato della flotta passa attraverso diverse fasi come Amazon GameLift Servers distribuisce i gruppi di container su ogni istanza della flotta e avvia i server di gioco. Quando la flotta raggiunge lo stato`ACTIVE`, è pronta per ospitare sessioni di gioco. Se crei una flotta senza una definizione di gruppo di container, la flotta non raggiungerà lo stato attivo. Per assistenza in caso di problemi con la creazione del parco istanze, consulta Debug dei problemi del parco istanze di Amazon GameLift Servers.

Console

Nella Amazon GameLift Servers console, seleziona Regione AWS dove vuoi creare la flotta. Le definizioni dei gruppi di contenitori devono trovarsi nella stessa regione in cui si desidera creare la flotta.

Apri la barra di navigazione a sinistra della console e scegli Contenitori gestiti: flotte. Nella pagina Flotte, scegli Crea flotta di container.

Fase 1: Definisci i dettagli della flotta di container gestita

1. Nella sezione Dettagli della flotta di container, inserisci una descrizione della flotta.
2. Specificate un ruolo IAM per la flotta. Questo ruolo dispone di autorizzazioni che Amazon GameLift Servers devi gestire la flotta di container per tuo conto. Per informazioni sulla creazione del ruolo di servizio richiesto, consultaConfigurare un ruolo di servizio IAM per Amazon GameLift Servers.
3. Scegliete un'opzione di configurazione del registro. L' CloudWatch opzione è selezionata per impostazione predefinita. Fornisci le informazioni richieste in base all'opzione selezionata.
4. Aggiungi gruppi di container alla flotta. Questa fase è opzionale. Puoi scegliere di creare una flotta senza un gruppo di container con un piano per aggiungerli in un secondo momento.

Una flotta senza gruppi di container non distribuirà alcuna istanza della flotta e non può ancora ospitare giochi, ma la risorsa della flotta viene creata.

- Seleziona una definizione del gruppo di contenitori del server di gioco. Facoltativamente, specifica la versione della definizione che desideri distribuire. Se non specifichi il numero di versione, Amazon GameLift Servers utilizza automaticamente la versione più recente.
- Facoltativamente, aggiungi una definizione e una versione del gruppo di contenitori per istanza. Se non specifichi il numero di versione, Amazon GameLift Servers utilizza automaticamente la versione più recente.

5. Nella sezione Dettagli aggiuntivi, puoi impostare alcune personalizzazioni opzionali. Nessuna di queste impostazioni è richiesta per creare la flotta di container.

Fase 2: Definizione dei dettagli dell'istanza

1. In Distribuzione delle istanze, seleziona una o più postazioni remote in cui distribuire le istanze. La regione di origine viene selezionata automaticamente (è la regione in cui stai creando la flotta). Se selezioni sedi aggiuntive, le istanze della flotta vengono distribuite anche in queste località.

   > **Important**
   >
   > Per utilizzare le regioni che non sono abilitate per impostazione predefinita, abilitale nelle tue. Account AWS
   >
   > - Le flotte con regioni non abilitate che hai creato prima del 28 febbraio 2022 non sono interessate.
   > - Per creare nuove flotte con più sedi o aggiornare le flotte con più sedi esistenti, abilita innanzitutto tutte le regioni che scegli di utilizzare.
   >
   > Per ulteriori informazioni sulle regioni che non sono abilitate per impostazione predefinita e su come abilitarle, consulta Gestione in. Regioni AWSRiferimenti generali di AWS

2. Seleziona una configurazione di istanza per il parco istanze. La console calcola automaticamente la vCPU e la memoria minime richieste (in base ai limiti totali impostati per ogni gruppo di contenitori). Filtra l'elenco completo dei tipi di istanze disponibili in base ai requisiti di risorse e alle posizioni inserite. Se necessario, puoi aggiungere filtri aggiuntivi.

Per ulteriori informazioni sulla scelta del tipo di istanza, consultaConfigura una flotta di container. La dimensione del tipo di istanza scelto influirà sul modo in cui i gruppi di container dei server di gioco vengono impacchettati in ciascuna istanza della flotta. A seconda della scelta, valuta la possibilità di rivedere le impostazioni relative ai gruppi di contenitori dei server di gioco desiderati per ogni istanza.

Fase 4: Revisione e creazione

- Rivedi le impostazioni di configurazione del tuo parco veicoli.

  Puoi aggiornare i metadati e la configurazione della flotta in qualsiasi momento, indipendentemente dallo stato della flotta. Per ulteriori informazioni, consulta Aggiornamento di una configurazione del Amazon GameLift Servers parco veicoli. Puoi aggiornare la capacità della flotta dopo che la flotta ha raggiunto lo stato ATTIVO. Per ulteriori informazioni, consulta Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers. Puoi anche aggiungere o rimuovere postazioni remote.

  Al termine della revisione, scegli Crea.

  Se la richiesta ha esito positivo, la console visualizza la pagina di dettaglio della nuova risorsa del parco veicoli. Inizialmente lo stato è `NEW` Amazon GameLift Servers avvia il processo di creazione della flotta. È possibile monitorare lo stato del nuovo parco istanze nella pagina Fleets (Parchi istanze). Una flotta è pronta per ospitare sessioni di gioco quando raggiunge lo statoACTIVE.

AWS CLI

Per creare una flotta di container con AWS CLI, apri una finestra della riga di comando e usa il `create-container-fleet` comando. Per ulteriori informazioni su questo comando, vedere `create-container-fleet`nella Guida ai AWS CLI comandi.

La `create-container-fleet` richiesta di esempio mostrata di seguito crea una nuova flotta di container con le seguenti caratteristiche:

- ContainerGroupsConfiguration Specifica solo una definizione di gruppo di contenitori per server di gioco:`MyAdventureGameContainerGroup`. Il numero di gruppi di contenitori di server di gioco che verranno distribuiti su ciascuna istanza della flotta viene calcolato da Amazon GameLift Servers.

- Per impostazione predefinita, la flotta utilizza istanze c5.large On-Demand.
- Per impostazione predefinita, la flotta apre una serie di porte di connessione e porte di autorizzazione in entrata, come calcolato da Amazon GameLift Servers. Distribuisce gruppi di contenitori nelle seguenti posizioni:

```
aws gamelift create-container-fleet \
    --fleet-role-arn arn:aws:iam::MyAccount:role/MyContainersRole \
    --game-server-container-group-definition-name "rn:aws:gamelift:us-
west-2:111122223333:containergroupdefinition/MyAdventureGameContainerGroup:2" \
```

Se la richiesta di creazione della flotta ha esito positivo, Amazon GameLift Servers restituisce un set di attributi della flotta che include le impostazioni di configurazione richieste e un nuovo ID della flotta di container. Amazon GameLift Servers quindi imposta lo stato della flotta e lo stato dell'ubicazione su Nuovo e avvia il processo di attivazione della flotta. Puoi monitorare lo stato del parco istanze e visualizzare altre informazioni tramite questi comandi dell'interfaccia CLI:

- describe-fleet-events
- describe-container-fleet
- describe-fleet-capacity
- describe-fleet-port-settings
- describe-fleet-utilization
- describe-fleet-location-capacity
- describe-fleet-location-utilization

È possibile modificarne la capacità e altre impostazioni di configurazione in base alle esigenze tramite questi comandi:

- update-container-fleet
- update-fleet-capacity
- update-fleet-port-settings
- create-fleet-locations
- delete-fleet-locations

Aggiorna un Amazon GameLift Servers flotta di container gestita

È possibile aggiornare la maggior parte delle proprietà di una flotta di container gestita, incluse le definizioni dei gruppi di container. A seconda delle impostazioni da aggiornare, un aggiornamento del parco veicoli potrebbe avviare una nuova implementazione del parco veicoli. In una distribuzione con un parco istanze, tutte le istanze del parco istanze vengono rimosse e sostituite con istanze con la nuova configurazione. Le impostazioni che richiedono una distribuzione includono:

- Definizioni dei gruppi di contenitori, inclusi gli aggiornamenti alle immagini dei contenitori
- Intervalli di porte di connessione e autorizzazioni in ingresso
- Configurazione del registro

È possibile tenere traccia dello stato delle implementazioni della flotta nel Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI) per creare una flotta di container.

Console

Nella Amazon GameLift Servers console, seleziona Regione AWS dove vuoi creare la flotta. Le definizioni dei gruppi di contenitori devono trovarsi nella stessa regione in cui si desidera creare la flotta.

Apri la barra di navigazione a sinistra della console e scegli Contenitori gestiti: flotte. Nella pagina Flotte di container gestite, seleziona una flotta dall'elenco e scegli Modifica.

1. Aggiorna le impostazioni della flotta di container secondo necessità. Al termine, scegliere Create (Crea).
2. Se gli aggiornamenti richiedono l'implementazione di una flotta, ti viene chiesto di specificare le opzioni di distribuzione come segue:

   - Protezione delle sessioni di gioco. Puoi scegliere di proteggere le istanze della flotta con sessioni di gioco attive (distribuzione sicura). Con questa impostazione, le istanze della flotta vengono sostituite solo dopo la fine delle sessioni di gioco. In alternativa, puoi scegliere di sostituire le istanze della flotta indipendentemente dall'attività della sessione di gioco (distribuzione non sicura). Le implementazioni non sicure sono utili durante le fasi di sviluppo e test per ridurre i tempi di implementazione.
   - Percentuale minima di salute. Puoi gestire la velocità con cui le istanze del parco istanze vengono sostituite. Utilizza questa impostazione per mantenere un numero minimo di attività integre durante la distribuzione. Un valore basso dà priorità alla velocità di

distribuzione, mentre un valore alto assicura che la disponibilità del server di gioco rimanga elevata per tutta la durata della distribuzione.

- Strategia di fallimento dell'implementazione. Decidi quali azioni intraprendere se una distribuzione fallisce. Un errore di distribuzione significa che alcuni contenitori aggiornati non hanno superato i controlli di stato e sono considerati compromessi. È possibile impostare le distribuzioni per ripristinare automaticamente tutte le istanze del parco istanze allo stato di distribuzione precedente. In alternativa, puoi scegliere di mantenere alcune delle istanze del parco istanze danneggiate da utilizzare nel debug.

Se la richiesta ha esito positivo, la console visualizza la scheda Distribuzioni per la flotta di container gestita. Utilizza questa scheda per tenere traccia dello stato di ogni implementazione. Se avvii una nuova distribuzione per la flotta, questa azione annulla automaticamente qualsiasi distribuzione attualmente in corso per la flotta.

AWS CLI

Per creare una flotta di container con AWS CLI, apri una finestra della riga di comando e usa il `update-container-fleet` comando. Per ulteriori informazioni su questo comando, vedere `update-container-fleet`nella Guida ai AWS CLI comandi.

L'esempio seguente aggiorna una flotta di container esistente con le seguenti caratteristiche:

- Aggiorna la definizione del gruppo di contenitori del server di gioco per utilizzare la versione 2.
- Specifica le opzioni di distribuzione sicure.

```
{
  "DeploymentConfiguration": {
    "ImpairmentStrategy": "ROLLBACK",
    "MinimumHealthyPercentage": 75,
    "ProtectionStrategy": "WITH_PROTECTION"
  },
  "FleetId": "containerfleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
  "GameServerContainerGroupDefinitionName": "arn:aws:gamelift:us-
west-2:111122223333:containergroupdefinition/MyAdventureGameContainerGroup:2"
}
```

Personalizza una flotta di Amazon GameLift Servers container

Gli argomenti di questa sezione descrivono alcune delle funzionalità opzionali per i contenitori Amazon GameLift Servers gestiti. Puoi scegliere di utilizzare una o tutte queste funzionalità.

Argomenti

Imposta i limiti delle risorse

Per ogni gruppo di contenitori, puoi determinare la quantità di memoria e potenza di calcolo di cui il gruppo di contenitori ha bisogno per eseguire il software. Amazon GameLift Serverssi basa su queste informazioni per gestire le risorse all'interno del gruppo di contenitori. Inoltre, utilizza queste informazioni per calcolare quanti gruppi di container di server di gioco può contenere un'immagine della flotta. Puoi anche impostare limiti per singoli contenitori.

È possibile impostare un limite massimo di memoria e potenza di calcolo per un gruppo di contenitori. Per impostazione predefinita, queste risorse sono condivise da tutti i contenitori del gruppo. È possibile personalizzare ulteriormente la gestione delle risorse impostando limiti per i singoli contenitori.

Imposta limiti opzionali per i singoli contenitori

> L'impostazione di limiti di risorse specifici per i contenitori consente di esercitare un maggiore controllo su come i singoli contenitori possono utilizzare le risorse del gruppo. Se non imposti limiti specifici del contenitore, tutti i contenitori del gruppo condividono le risorse del gruppo. La condivisione offre una maggiore flessibilità nell'utilizzo delle risorse laddove sono necessarie. Inoltre, aumenta la possibilità che i processi competano tra loro e causino il fallimento dei container.
>
> Imposta una delle seguenti `ContainerDefinition` proprietà per qualsiasi contenitore.

- `MemoryHardLimitMebibytes`— Imposta un limite massimo di memoria per il contenitore. Se il contenitore supera questo limite, si verifica un riavvio.
- `Vcp`ulimit: riserva una quantità minima di risorse vCPU per l'uso esclusivo del contenitore. Il contenitore ha sempre a disposizione la quantità riservata. Può superare questo minimo in qualsiasi momento, se sono disponibili risorse aggiuntive. (1024 unità CPU equivalgono a 1 vCPU.)

Imposta i limiti totali delle risorse per un gruppo di contenitori

Se si impostano limiti per singoli contenitori, potrebbe essere necessario modificare la quantità di memoria e di risorse vCPU necessarie al gruppo di contenitori. L'obiettivo è allocare risorse sufficienti per ottimizzare le prestazioni del server di gioco. Amazon GameLift Serversutilizza questi limiti per calcolare come impacchettare i gruppi di container dei server di gioco su un'istanza di fleet. Li utilizzerai anche per scegliere un tipo di istanza per una flotta di container.

Calcola la memoria totale e la vCPU necessarie per un gruppo di contenitori. Considera i seguenti aspetti:

- Quali sono tutti i processi che vengono eseguiti su tutti i contenitori del gruppo di contenitori? Somma le risorse necessarie per questi processi. Prendi nota di eventuali limiti specifici del contenitore.
- Quanti processi simultanei dei server di gioco intendi eseguire in ciascun gruppo di container? Lo determini nell'immagine del contenitore del server di gioco.

In base alla stima dei requisiti del gruppo di contenitori, imposta le seguenti `ContainerGroupDefinition` proprietà:

- `TotalMemoryLimitMebibytes`— Imposta un limite massimo di memoria per il gruppo di contenitori. Tutti i contenitori del gruppo condividono la memoria allocata. Se si impostano limiti per singoli contenitori, il limite di memoria totale deve essere uguale o superiore al limite di memoria specifico del contenitore più elevato.
- `TotalVcpuLimit`— Imposta un limite massimo di vCPU per il gruppo di contenitori. Tutti i contenitori del gruppo condividono le risorse CPU allocate. Se imposti limiti dei singoli contenitori, il limite totale della CPU deve essere uguale o superiore alla somma di tutti i limiti della CPU specifici del contenitore. Come procedura ottimale, valuta la possibilità di impostare questo valore per raddoppiare la somma dei limiti della CPU del contenitore.

Scenario di esempio

Supponiamo di definire un gruppo di contenitori di server di gioco con i seguenti tre contenitori:

- Il contenitore A è il nostro contenitore per server di gioco. Stimiamo i requisiti di risorse per un server di gioco a 512 MiB e 1024 CPU. Prevediamo che il contenitore esegua 1 processo server. Poiché questo contenitore esegue il nostro software più importante, non abbiamo impostato alcun limite di memoria o limite di riserva per vCPU.
- Il container B run è un contenitore di supporto con requisiti di risorse stimati in 1024 MiB e 1536 CPU. Abbiamo impostato un limite di memoria di 2048 MiB e un limite di riserva della CPU di 1024 CPU.
- Il contenitore C è un altro contenitore di supporto. Abbiamo impostato un limite di memoria rigida di 512 MiB e un limite di riserva della CPU di 512 CPU.

Utilizzando queste informazioni, impostiamo i seguenti limiti totali per il gruppo di contenitori:

- Limite di memoria totale: 7680 MiB. Questo valore supera il limite di memoria massimo (1024 MiB).
- Limite totale della CPU: 13312 CPU. Questo valore supera la somma del limite della CPU (1024+512 CPU).

Designare contenitori essenziali

Per un gruppo di contenitori per istanza, designa ogni contenitore come essenziale o non essenziale. I gruppi di contenitori per istanza devono avere almeno un contenitore di supporto essenziale. Il contenitore essenziale svolge il lavoro fondamentale del gruppo di contenitori. Si prevede che il contenitore essenziale sia sempre in esecuzione. Se fallisce, l'intero gruppo di contenitori si riavvia.

Imposta la `ContainerDefinition` proprietà su `Essential` true o false per ogni contenitore.

Configurare le connessioni di rete

Puoi personalizzare l'accesso alla rete per consentire al traffico esterno di connettersi a qualsiasi container di una flotta di container. Ad esempio, devi stabilire connessioni di rete al contenitore che esegue i processi del server di gioco, in modo che i client di gioco possano unirsi e giocare al gioco. I client di gioco si connettono ai server di gioco utilizzando porte e indirizzi IP.

In una flotta di container, la connessione tra client e server non è diretta. Internamente, un processo in un container ascolta su una porta container. Esternamente, il traffico in entrata si connette a un'istanza della flotta utilizzando una porta di connessione. Amazon GameLift Serversmantiene le mappature tra le porte container interne e le porte di connessione rivolte verso l'esterno, in modo che il traffico in entrata venga indirizzato al processo corretto sull'istanza.

Amazon GameLift Serversfornisce un ulteriore livello di controllo per le connessioni di rete. Ogni flotta di container dispone di un'impostazione delle autorizzazioni in entrata, che consente di controllare l'accesso a ciascuna porta di connessione rivolta verso l'esterno. Ad esempio, è possibile rimuovere le autorizzazioni per tutte le porte di connessione per impedire l'accesso ai container della flotta.

Puoi aggiornare le autorizzazioni in entrata, le porte di connessione e le porte container di una flotta.

Imposta gli intervalli di porte del contenitore

Configura gli intervalli di porte del contenitore come parte della definizione di ogni contenitore. Questo è un parametro obbligatorio per la definizione di un gruppo di contenitori. È necessario configurare un numero sufficiente di porte per ospitare tutti i processi in esecuzione simultanea che richiedono l'accesso esterno. Alcuni contenitori non avranno bisogno di porte.

Il contenitore del server di gioco, che gestisce i server di gioco, necessita di una porta per ogni processo del server di gioco in esecuzione contemporaneamente. Il processo del server di gioco ascolta sulla porta assegnata e lo segnala a. Amazon GameLift Servers

Imposta gli intervalli delle porte di connessione

Configura la tua flotta di container con un set di porte di connessione. Le porte di connessione forniscono l'accesso esterno alle istanze del parco istanze che eseguono i container. Amazon GameLift Serversassegna le porte di connessione e le mappa alle porte dei container in base alle esigenze.

Per impostazione predefinita, Amazon GameLift Servers calcola il numero di porte necessarie per tutti i gruppi di container e imposta un intervallo di porte per soddisfarle. Ti consigliamo vivamente di utilizzare valori Amazon GameLift Servers calcolati, che vengono aggiornati quando distribuisci gli aggiornamenti a una definizione di gruppo di contenitori. Se è necessario personalizzare gli intervalli di porte di connessione, utilizzare le seguenti indicazioni.

Quando crei una flotta di container, definisci un intervallo di porte di connessione (vedi ContainerFleet: InstanceConnectionPortRange). Assicurati che l'intervallo contenga un numero sufficiente di porte da mappare a ogni porta container definita in tutti i container di entrambi i gruppi di container della flotta. Per calcolare le porte di connessione minime necessarie, utilizza la seguente formula:

```
[Total number of container ports defined for containers in the game
server container group] * [Number of game server container groups per
instance] + [Total number of container ports defined for containers in
the per-instance container group]
```

È consigliabile raddoppiare il numero minimo di porte di connessione.

> **Note**
>
> Il numero di porte di connessione può potenzialmente limitare il numero di gruppi di contenitori di server di gioco per istanza. Se una flotta dispone di un numero sufficiente di porte di connessione per un gruppo di container di server di gioco per istanza, Amazon GameLift Servers implementerà un solo gruppo di contenitori di server di gioco, anche se le istanze dispongono di una potenza di calcolo sufficiente per più gruppi di contenitori di server di gioco.

Imposta le autorizzazioni in entrata

Le autorizzazioni in entrata controllano l'accesso esterno a una flotta di container specificando quali porte di connessione aprire per il traffico in entrata. Puoi utilizzare questa impostazione per attivare e disattivare l'accesso alla rete di una flotta in base alle esigenze.

Per impostazione predefinita, Amazon GameLift Servers calcola il numero di porte necessarie per tutti i gruppi di container e imposta un intervallo di porte per soddisfarle. Ti consigliamo vivamente di utilizzare valori Amazon GameLift Servers calcolati, che vengono aggiornati quando distribuisci gli aggiornamenti a una definizione di gruppo di contenitori. Se è necessario personalizzare gli intervalli di porte di connessione, utilizzare le seguenti indicazioni.

Quando crei una flotta di container, definisci un set di autorizzazioni in entrata (vedi ContainerFleet: InstanceInboundPermisssions). Le porte di autorizzazione in entrata devono corrispondere agli intervalli di porte di connessione della flotta.

Scenario di esempio

Questo esempio illustra come impostare tutte e tre le proprietà della connessione di rete.

- Il gruppo di container di server di gioco della nostra flotta dispone di 1 container, che esegue i processi di 1 server di gioco.

  Nella definizione del gruppo di contenitori del server di gioco, impostiamo il `PortConfiguration` parametro per questo contenitore come segue:

```
"PortConfiguration": {
  "ContainerPortRanges": [ { "FromPort": 10, "ToPort": 20, "Protocol": "TCP"} ]
  }
```

- La nostra flotta dispone anche di un gruppo di container per istanza con 1 container. Ha 1 processo che richiede l'accesso alla rete. Nella definizione del contenitore per istanza, impostiamo il `PortConfiguration` parametro per questo contenitore come segue:

  ```
  "PortConfiguration": {
    "ContainerPortRanges": [ { "FromPort": 25, "ToPort": 25, "Protocol": "TCP"} ]  }
  ```

- La nostra flotta è configurata con 20 gruppi di container di server di gioco per istanza di flotta. Alla luce di queste informazioni, possiamo usare la formula per calcolare il numero di porte di connessione di cui abbiamo bisogno:
  - Minimo: 21 porte [1 porta contenitore per server di gioco * 20 gruppi di contenitori di server di gioco per istanza + 1 porta container per istanza]
  - Migliore pratica: 42 porte [porte minime* 2]

  Durante la creazione della flotta di container, impostiamo il `ConnectionPortRange` parametro come segue:

  ```
  "ConnectionPortRange": { "FromPort": 1010, "ToPort": 1071 }
  ```

- Vogliamo consentire l'accesso a tutte le porte di connessione disponibili. Durante la creazione della flotta di container, impostiamo il `InstanceInboundPermissions` parametro come segue:

  ```
  "InstanceInboundPermissions": [
    {"FromPort": 1010, "ToPort": 1071, "IpRange": "10.24.34.0/23", "Protocol":
   "TCP"} ]
  ```

Imposta i controlli sanitari per i container

Un contenitore si riavvia automaticamente se si verifica un errore del terminale e smette di funzionare. Se un contenitore è considerato essenziale, richiede il riavvio dell'intero gruppo di contenitori.

Tutti i contenitori del server di gioco vengono automaticamente considerati essenziali. I container di supporto possono essere considerati essenziali, ma devono disporre di un meccanismo per segnalare lo stato. Puoi impostare controlli di integrità anche per i contenitori di supporto non essenziali.

È possibile definire criteri personalizzati aggiuntivi per misurare lo stato dei contenitori e utilizzare un controllo dello stato per verificare tali criteri. Per impostare un controllo dello stato del contenitore, puoi definirlo in un'immagine del contenitore Docker o nella definizione del contenitore. Se imposti un controllo dello stato nella definizione del contenitore, questo sovrascrive qualsiasi impostazione nell'immagine del contenitore.

Imposta le seguenti `SupportContainerDefinition` proprietà per il controllo dello stato del contenitore:

- `Command`— Fornisci un comando che controlli alcuni aspetti dello stato del contenitore. Siete voi a decidere quali criteri utilizzare per misurare lo stato di salute. Il comando deve restituire un valore di uscita pari a 1 (non salutare) o 0 (sano).
- `StartPeriod`— Specificare un ritardo iniziale prima che gli errori del controllo sanitario inizino a contare. Questo ritardo dà al contenitore il tempo di avviare i propri processi.
- `Interval`— Decidi con che frequenza eseguire il comando health check. Con quale rapidità desiderate rilevare e risolvere un guasto del contenitore?
- `Timeout`— Decidi per quanto tempo attendere l'esito positivo o negativo prima di riprovare il comando health check. Quanto tempo deve impiegare per il completamento del comando di controllo dello stato di salute?
- `Retries`— Quante volte è necessario riprovare il comando health check prima di registrare un errore?

Imposta le dipendenze del contenitore

All'interno di ogni gruppo di contenitori è possibile impostare le dipendenze tra i contenitori in base allo stato del contenitore. Una dipendenza influisce sul momento in cui il contenitore dipendente può avviarsi o chiudersi in base allo stato di un altro contenitore.

Un caso d'uso chiave per le dipendenze consiste nella creazione di sequenze di avvio e chiusura per il gruppo di contenitori.

Ad esempio, potresti volere che il contenitore A venga avviato per primo e venga completato correttamente prima dell'avvio dei contenitori B e C. A tale scopo, create innanzitutto una dipendenza per il contenitore B dal contenitore A, a condizione che il contenitore A venga completato correttamente. Quindi crea una dipendenza per il contenitore C sul contenitore A con la stessa condizione. Le sequenze di avvio si verificano in ordine inverso rispetto all'arresto.

Configura una flotta di container

Quando crei una flotta di container, considera i seguenti punti decisionali. La maggior parte di questi punti dipende dall'architettura e dalla configurazione del container.

Decidi dove vuoi distribuire la tua flotta

In generale, vuoi schierare le tue flotte geograficamente vicino ai giocatori per ridurre al minimo la latenza. Puoi distribuire la tua flotta di container su qualsiasi dispositivo Each che supporti. Regione AWS Amazon GameLift Servers Se desideri distribuire lo stesso server di gioco in altre località geografiche, puoi aggiungere località remote alla flotta, tra cui Regioni AWS Local Zones. Per una flotta con più sedi, puoi regolare la capacità in modo indipendente in ogni sede del parco veicoli. Per ulteriori informazioni sulle sedi del parco veicoli supportate, consultaAmazon GameLift Serverssedi di assistenza.

Scegli un tipo e una dimensione di istanza per il tuo parco istanze

Amazon GameLift Serverssupporta un'ampia gamma di tipi di EC2 istanze Amazon, tutte disponibili per l'uso con una flotta di container. La disponibilità e il prezzo del tipo di istanza variano in base alla località. È possibile visualizzare un elenco dei tipi di istanze supportati, filtrati per posizione, nella Amazon GameLift Servers console (in Risorse, Istanze e quote di servizio).

Quando scegli un tipo di istanza, considera innanzitutto la famiglia di istanze. Le famiglie di istanze offrono varie combinazioni di CPU, memoria, archiviazione e funzionalità di rete. Ottieni maggiori informazioni sulle famiglie di EC2 istanze. All'interno di ogni famiglia hai diverse dimensioni di istanze tra cui scegliere. Considerate i seguenti aspetti quando selezionate la dimensione di un'istanza:

- Qual è la dimensione minima dell'istanza in grado di supportare il tuo carico di lavoro? Utilizza queste informazioni per eliminare i tipi di istanza troppo piccoli.
- Quali sono le dimensioni dei tipi di istanza più adatte alla tua architettura di container? Idealmente, dovresti scegliere una dimensione che possa ospitare più copie del gruppo di contenitori del tuo server di gioco con uno spreco minimo di spazio.
- Quale granularità di scalabilità è utile per il tuo gioco? La capacità di Scale Fleet prevede l'aggiunta o la rimozione di istanze e ogni istanza rappresenta la capacità di ospitare un numero specifico di sessioni di gioco. Considera quanta capacità desideri aggiungere o rimuovere con ogni istanza. Se la domanda dei giocatori varia di migliaia di minuti in minuto, potrebbe essere opportuno utilizzare istanze molto grandi in grado di ospitare centinaia o migliaia di sessioni di gioco. Al contrario, potresti preferire un controllo di ridimensionamento più preciso con tipi di istanze più piccoli.

- Sono disponibili risparmi sui costi in base alle dimensioni? Potresti scoprire che il costo di alcuni tipi di istanze varia in base alla località a causa della disponibilità.

Imposta altre impostazioni opzionali del parco veicoli

Puoi utilizzare le seguenti funzionalità opzionali durante la configurazione di una flotta di container:

- Configura i server di gioco per accedere ad altre AWS risorse. Consultare Comunica con altre AWS risorse delle tue flotte.
- Proteggi le sessioni di gioco con giocatori attivi dalle interruzioni premature durante un evento a scala ridotta.
- Limita il numero di sessioni di gioco che un individuo può creare sulla flotta entro un periodo di tempo limitato.

Creare una definizione di gruppo di container per una flotta di Amazon GameLift Servers container

Una definizione di gruppo di container descrive come distribuire le applicazioni di server di gioco containerizzate in una flotta di container. È un modello che indica Amazon GameLift Servers quali immagini di container distribuire nella flotta e come eseguirle. Quando crei una flotta di container, specifichi le definizioni dei gruppi di container da distribuire nella flotta. Per ulteriori informazioni sui gruppi di container, consultaComponenti della flotta di container.

Prima di iniziare

Suggerimenti su cosa fare prima di iniziare a creare una definizione di gruppo di contenitori:

- Finalizza le immagini dei container e trasferiscile in un repository Amazon Elastic Container Registry (Amazon ECR) nello stesso in Regione AWS cui intendi creare il gruppo di contenitori. Amazon GameLift Serversacquisisce un'istantanea di ogni immagine al momento della creazione della definizione del gruppo di contenitori e la utilizza durante la distribuzione in una flotta di container. Consultare Crea un'immagine del contenitore per Amazon GameLift Servers.
- Crea le definizioni dei contenitori come file JSON. Una definizione di gruppo di contenitori include una o più definizioni di contenitori. È possibile utilizzare i file JSON se si crea una definizione di gruppo di contenitori utilizzando. AWS CLIfor Amazon GameLift Servers
- Verifica che il tuo AWS utente disponga delle autorizzazioni IAM per accedere al repository Amazon ECR. Consultare Esempi di autorizzazioni IAM per Amazon GameLift Servers.

Crea una definizione di gruppo di contenitori per server di gioco

Un gruppo di contenitori di server di gioco esegue il software del server di gioco. Un gruppo di contenitori di server di gioco ha un contenitore di server di gioco, che esegue l'eseguibile del server di gioco. Può anche avere uno o più contenitori di supporto per eseguire software aggiuntivo a supporto del server di gioco. (A volte vengono definiti contenitori «sidecar».)

Questo argomento descrive come creare una semplice definizione di gruppo di contenitori di server di gioco utilizzando la Amazon GameLift Servers console o gli AWS strumenti CLI. Per informazioni più dettagliate sulle funzionalità opzionali, consultaPersonalizza una flotta di Amazon GameLift Servers container.

> **Note**
>
> È possibile modificare la maggior parte delle definizioni e delle impostazioni dei gruppi di contenitori dopo averle create. Se apporti modifiche alla definizione di un contenitore, Amazon GameLift Servers acquisisce una nuova istantanea delle immagini del contenitore aggiornate.

Per creare una semplice definizione di gruppo di contenitori per server di gioco:

Le seguenti istruzioni descrivono come creare una definizione di gruppo di contenitori con i parametri minimi richiesti e utilizzando i valori Amazon GameLift Servers predefiniti.

Console

Nella Amazon GameLift Serversconsole, seleziona il Regione AWS punto in cui desideri creare il gruppo di contenitori.

Apri la barra di navigazione a sinistra della console e scegli Contenitori gestiti: definizioni di gruppo. Nella pagina di definizione dei gruppi di contenitori, scegli Crea definizione di gruppo.

Fase 1: Definire i dettagli della definizione del gruppo di contenitori

1. Immettere un nome per la definizione del gruppo di contenitori. Il nome deve essere univoco per la regione Account AWS and.
2. Seleziona il tipo di gruppo di contenitori del server di gioco.

3. Per Limite di memoria totale, inserisci le risorse di memoria massime da rendere disponibili per tutti i contenitori del gruppo di contenitori. Per informazioni sul calcolo di questo valore, consultaImposta i limiti delle risorse.
4. Per Limite totale di vCPU, inserisci la potenza di calcolo massima da rendere disponibile per tutti i contenitori del gruppo di contenitori. Per informazioni sul calcolo di questo valore, consulta. Imposta i limiti delle risorse

Fase 2: Aggiungere le definizioni dei contenitori

Un gruppo di contenitori di server di gioco ha almeno un contenitore di server di gioco. Nella console, la prima definizione di contenitore che crei è il contenitore del server di gioco. Questo passaggio descrive come definire le impostazioni minime richieste per la definizione di un contenitore del server di gioco.

1. Inserisci il nome della definizione del contenitore. Ogni contenitore definito per il gruppo deve avere un valore di nome univoco.
2. Collegati a un'immagine del contenitore con la build del tuo server di gioco. Inserisci l'URI dell'immagine Amazon ECR per l'immagine di un contenitore in un repository pubblico o privato. Puoi utilizzare uno dei seguenti formati:
   - Solo URI dell'immagine: `[Account AWS].dkr.ecr.[Regione AWS].amazonaws.com/[repository ID]`
   - URI dell'immagine + digest: `[Account AWS].dkr.ecr.[Regione AWS].amazonaws.com/[repository ID]@[digest]`
   - Tag URI + dell'immagine: `[Account AWS].dkr.ecr.[Regione AWS].amazonaws.com/[repository ID]:[tag]`
3. Specificate la versione Amazon GameLift Servers Server SDK utilizzata dalla build del server di gioco. Per una flotta di container, questo valore deve essere 5.2.0 o superiore.
4. In Intervallo di porte del container interno, imposta il protocollo e definisci un intervallo di porte. La dimensione dell'intervallo deve essere maggiore del numero di processi simultanei del server di gioco che verranno eseguiti in questo contenitore. Se il contenitore del server di gioco esegue solo un processo server per contenitore, questo intervallo di porte richiede solo poche porte. Per ulteriori dettagli, consulta Configurare le connessioni di rete.
5. Aggiungi altri contenitori se necessario per eseguire software di supporto aggiuntivo. I contenitori aggiuntivi vengono designati automaticamente come contenitori di supporto. Un

gruppo di contenitori di server di gioco può avere solo un contenitore di server di gioco e fino a otto contenitori di supporto. Fornisci le seguenti impostazioni minime richieste:

- Nome della definizione del contenitore
- URI dell'immagine ECR.
- Porte interne del contenitore (includetele solo se il contenitore ha processi che richiedono l'accesso alla rete).

Fase 3: Configurare le dipendenze

- Se la definizione del gruppo di contenitori include più di un contenitore, puoi facoltativamente impostare le dipendenze tra i contenitori. Per ulteriori informazioni, consulta Imposta le dipendenze del contenitore.

Fase 3: Revisione e creazione

1. Rivedi tutte le impostazioni di definizione del gruppo di contenitori. Usa Modifica per apportare modifiche a qualsiasi sezione, inclusa ciascuna delle definizioni dei contenitori per il gruppo.
2. Al termine della revisione, scegli Crea.

   Se la richiesta ha esito positivo, la console visualizza la pagina di dettaglio per la nuova risorsa di definizione del gruppo di contenitori. Inizialmente lo stato è`COPYING`: Amazon GameLift Servers inizia a scattare istantanee di tutte le immagini del contenitore per il gruppo. Al termine di questa fase, lo stato della definizione del gruppo di contenitori cambia in`READY`. Una definizione di gruppo di container deve essere `READY` attiva prima di poter creare una flotta di container con essa.

AWS CLI

Quando utilizzi la AWS CLI per creare una definizione di gruppo di contenitori, mantieni le configurazioni della definizione del contenitore in un file separato. `JSON` Puoi fare riferimento al file nel tuo comando CLI. Vedi alcuni esempi Crea un file di definizione del contenitore `JSON` di schema.

Crea una definizione di gruppo di contenitori

Per creare una nuova definizione di gruppo di contenitori, utilizzate il `create-container-group-definition` comando CLI. Per ulteriori informazioni su questo comando, [create-container-group-definition](#)consultate la AWS CLI Command Reference.

Questo esempio illustra una richiesta per la definizione di un gruppo di contenitori di server di gioco. Si presuppone che tu abbia creato un file JSON con le definizioni dei contenitori per questo gruppo.

```
aws gamelift create-container-group-definition \
    --name MyAdventureGameContainerGroup \
    --operating-system AMAZON_LINUX_2023 \
    --container-group-type GAME_SERVER \
    --total-memory-limit-mebibytes 4096 \
    --total-vcpu-limit 1 \
    --game-server-container-definition file://MyAdventureGameContainers.json
```

Crea un file di definizione del contenitore **JSON**

Quando si crea una definizione di gruppo di contenitori, si definiscono anche i contenitori per il gruppo. Una definizione di contenitore specifica il repository Amazon ECR in cui è archiviata l'immagine del contenitore e le configurazioni opzionali per le porte di rete, i limiti per l'utilizzo di CPU e memoria e altre impostazioni. Consigliamo di creare un singolo `JSON` file con le configurazioni per tutti i contenitori in un gruppo di contenitori. La manutenzione di un file è utile per archiviare, condividere e tenere traccia delle versioni di queste configurazioni critiche. Se si utilizza la AWS CLI per creare le definizioni dei gruppi di contenitori, è possibile fare riferimento al file nel comando.

Per creare una definizione di contenitore

1. Crea e apri un nuovo `.JSON` file. Per esempio:

   ```
   [~/work/glc]$ vim SimpleServer.json
   ```

2. Crea una definizione di contenitore separata per ciascuno dei contenitori del gruppo. Copia il seguente contenuto di esempio e modificalo secondo necessità per i tuoi contenitori. Per i dettagli sulla sintassi di una definizione di contenitore, consulta [ContainerDefinitionInput](#)l'Amazon GameLift ServersAPI Reference.
3. Salvate il file localmente in modo da potervi fare riferimento in un comando AWS CLI.

Esempio: definizione del contenitore del server di gioco

Example

Questo esempio descrive il contenitore essenziale per il gruppo di contenitori del server di gioco. Il contenitore di replica essenziale include l'applicazione per server di gioco, l'Amazon GameLift ServersAgent, e può includere altro software di supporto per l'hosting dei giochi. La definizione deve includere un nome, un URI dell'immagine e una configurazione di porta. Questo esempio imposta anche alcuni limiti di risorse specifici del contenitore.

```
  {
    "ContainerName": "MyAdventureGameServer",
    "ImageUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/gl-
containers:myadventuregame-server",
    "PortConfiguration": {
      "ContainerPortRanges": [
        {
          "FromPort": 2000,
          "Protocol": "TCP",
          "ToPort": 2010
        }
      ]
    },
    "ServerSdkVersion": "5.2.0"
  }
```

Aggiornare una definizione di gruppo di contenitori per un Amazon GameLift Servers flotta di container

È possibile aggiornare la maggior parte delle proprietà di una definizione di gruppo di contenitori, incluse le definizioni dei singoli contenitori. Le definizioni dei gruppi di contenitori hanno un numero di versione. Quando aggiorni una definizione di gruppo di contenitori, Amazon GameLift Servers salva l'aggiornamento e incrementa il numero di versione della definizione. Quando si configura una flotta di container, è possibile specificare quale versione della definizione di gruppo di contenitori distribuire.

Dopo aver aggiornato la definizione di un gruppo di container, puoi distribuire la nuova versione su una flotta di container nuova o esistente.

Aggiorna la definizione del gruppo di contenitori di un server di gioco

Questo argomento descrive come aggiornare la definizione del gruppo di contenitori del server di gioco utilizzando il Amazon GameLift Servers strumenti di console o AWS CLI. Per informazioni più dettagliate sulle funzionalità opzionali, vedere[Personalizza una flotta di Amazon GameLift Servers container](#).

Per aggiornare una definizione di gruppo di contenitori:

Console

Nella Amazon GameLift Servers console, seleziona il Regione AWS punto in cui desideri creare il gruppo di contenitori.

Apri la barra di navigazione a sinistra della console e scegli Contenitori gestiti: definizioni di gruppo. Nella pagina di definizione dei gruppi di contenitori, scegli una definizione e una versione del gruppo di contenitori da aggiornare.

Dopo aver salvato gli aggiornamenti, puoi utilizzare la nuova versione per creare nuove flotte di container oppure puoi distribuire gli aggiornamenti a una flotta di container esistente.

Fase 1: Definizione dei dettagli della definizione del gruppo di contenitori

- È possibile aggiornare le impostazioni dei limiti di memoria totale e vCPU.

Fase 2: Aggiungere le definizioni dei contenitori

È possibile apportare i seguenti aggiornamenti delle definizioni dei contenitori:

- Aggiorna le definizioni dei contenitori esistenti.
- Aggiungi nuove definizioni dei contenitori di supporto.
- Rimuovi le definizioni dei contenitori di supporto.

1. È possibile aggiornare l'URI dell'immagine ECR. Assicurati di aggiornare l'impostazione della versione di Server SDK in modo che corrisponda alla nuova immagine.
2. È possibile aggiornare l'intervallo di porte interne del contenitore in base alle esigenze. Le modifiche apportate a queste impostazioni potrebbero influire sulle impostazioni delle porte di connessione di una flotta di container quando queste modifiche vengono implementate in una flotta. Per ulteriori dettagli, consulta Configurare le connessioni di rete.

Fase 3: Configurare le dipendenze

- È possibile modificare le dipendenze in base alle esigenze. Per ulteriori informazioni, consulta Imposta le dipendenze del contenitore.

Fase 3: Revisione e creazione

- Controlla gli aggiornamenti delle definizioni dei gruppi di contenitori. Usa Modifica per apportare ulteriori modifiche in qualsiasi sezione. Al termine, scegli Crea per generare una nuova versione della definizione del gruppo di contenitori.

  Se la richiesta ha esito positivo, la console visualizza la pagina di dettaglio per la nuova risorsa di definizione del gruppo di contenitori. Inizialmente lo stato è`COPYING`, come Amazon GameLift Servers inizia a scattare istantanee di tutte le immagini del contenitore per il gruppo. Al termine di questa fase, lo stato della definizione del gruppo di contenitori cambia in`READY`. Una definizione di gruppo di container deve essere `READY` attiva prima di poter creare una flotta di container con essa.

AWS CLI

Quando utilizzi la AWS CLI per creare o aggiornare una definizione di gruppo di contenitori, mantieni le configurazioni della definizione del contenitore in un file separato. `JSON` Puoi fare riferimento al file nel tuo comando CLI. Vedi esempi Crea un file di definizione del contenitore `JSON` di schemi.

Quando aggiorni una definizione, devi solo specificare i valori che desideri aggiornare. Amazon GameLift Servers conserva tutti i valori che non includi nella richiesta di aggiornamento. Se stai modificando la definizione di un contenitore. Tuttavia, quando modificate la definizione di un contenitore, fornite un set completo.

Per aggiornare una definizione di gruppo di contenitori

Per aggiornare una nuova definizione di gruppo di contenitori, utilizzate il `update-container-group-definition` comando CLI. Per ulteriori informazioni su questo comando, update-container-group-definitionconsultate la AWS CLI Command Reference.

Example : gruppo di contenitori per server di gioco

È possibile specificare una versione della definizione del gruppo di contenitori durante il recupero, l'aggiornamento o l'eliminazione di una definizione del gruppo di contenitori o durante la creazione o l'aggiornamento di una flotta di container. Ogni definizione di gruppo di contenitori ha una proprietà di versione. Inoltre, il valore ARN della definizione and specifica il numero di versione.

Questo esempio illustra una richiesta di modifica alla definizione di un gruppo di contenitori di server di gioco. Si presuppone che tu abbia creato un file JSON con le definizioni dei contenitori per questo gruppo. Questo esempio utilizza il valore ARN per il nome della definizione e specifica che l'aggiornamento è alla versione 1.

```
aws gamelift update-container-group-definition \
    --name arn:aws:gamelift:us-west-2:111122223333:containergroupdefinition/
MyAdventureGameContainerGroup:1 \
    --operating-system AMAZON_LINUX_2023 \
    --container-group-type GAME_SERVER \
    --total-memory-limit-mebibytes 4096 \
    --total-vcpu-limit 1 \
    --container-definitions file://SimpleServer.json
```

Clonare una definizione di gruppo di contenitori

Puoi utilizzare il plugin Amazon GameLift Servers console per clonare una definizione di gruppo di contenitori esistente.

Per clonare un gruppo di contenitori

1. Nella Amazon GameLift Servers console, vai al riquadro di navigazione a sinistra e scegli Gruppi di contenitori.
2. Nella pagina di elenco dei gruppi di contenitori, seleziona il gruppo di contenitori esistente che desideri clonare. Dopo aver selezionato un gruppo di contenitori, il pulsante Clona è attivo.
3. Seleziona Clona. Questa azione apre la procedura guidata per la creazione di gruppi di contenitori con impostazioni precompilate.
4. Immettete un nuovo nome per il gruppo di contenitori clonato. Il gruppo di contenitori nella stessa regione deve avere nomi univoci.

5. Scorri le pagine del gruppo di contenitori e delle definizioni dei contenitori, esamina e crea il nuovo gruppo di contenitori.

Eliminare una definizione di gruppo di contenitori per un Amazon GameLift Servers flotta di container

Sono disponibili diverse opzioni per eliminare una definizione di gruppo di contenitori. Quando si elimina una definizione di gruppo di contenitori, questa azione elimina anche tutte le definizioni di contenitori nel gruppo di contenitori.

Le definizioni dei gruppi di contenitori possono avere più versioni. Le versioni dei gruppi di contenitori hanno lo stesso nome, ma hanno un numero di versione diverso. La definizione del gruppo di contenitori ARNs specifica sia il nome che la versione.

È possibile specificare una versione della definizione del gruppo di contenitori durante il recupero, l'aggiornamento o l'eliminazione di una definizione di gruppo di contenitori o durante la creazione o l'aggiornamento di una flotta di contenitori. Ogni definizione di gruppo di contenitori ha una proprietà di versione. Inoltre, il valore ARN della definizione specifica il numero di versione.

Esistono diversi modi per eliminare le definizioni dei gruppi di contenitori:

- È possibile eliminare tutte le versioni di una definizione specifica.
- È possibile eliminare una versione particolare di una definizione specifica.
- È possibile conservare un certo numero di versioni più recenti ed eliminare le versioni precedenti di una definizione specifica. Ad esempio, è possibile eliminare tutte le versioni precedenti alla versione 5.

È possibile eliminare una versione della definizione del gruppo di contenitori solo se non viene utilizzata in una flotta di container.

Per eliminare una definizione di gruppo di contenitori

Console

Nella Amazon GameLift Servers console, seleziona il Regione AWS punto in cui desideri creare il gruppo di contenitori.

Apri la barra di navigazione a sinistra della console e scegli Contenitori gestiti: definizioni di gruppo. Nella pagina Definizioni dei gruppi di contenitori, seleziona la definizione che desideri modificare e scegli Elimina.

Ti viene richiesto di selezionare il tipo di eliminazione che desideri effettuare e di specificare impostazioni aggiuntive a seconda del tipo di eliminazione.

AWS CLI

- Per eliminare una definizione di gruppo di contenitori, utilizza il comando `delete-container-group-definition` CLI e fornisci i valori per il tipo di eliminazione che desideri eseguire. Per ulteriori informazioni su questo comando, delete-container-group-definitionconsultate la AWS CLI Command Reference.

  Questo esempio illustra una richiesta di eliminazione di tutte le versioni della definizione del gruppo di contenitori del server di gioco precedenti alla versione 5.

  Example

  ```
  aws gamelift delete-container-group-definition \
      --name MyAdventureGameContainerGroup \
      --version-count-to-retain 5 \
  ```

Amazon GameLift Servers Flotte ovunque

Utilizzo Anywhere flotte di cui vuoi approfittare Amazon GameLift Servers funzionalità con le tue risorse di hosting. Le flotte Anywhere vengono comunemente utilizzate come ambienti di test per lo sviluppo iterativo o insieme alle flotte gestite in una soluzione di hosting ibrida.

Una flotta Anywhere è costituita da un set di risorse di elaborazione (virtuali o fisiche) fornite e gestite dall'utente. I computer possono risiedere in qualsiasi posizione geografica con connettività, da un laptop locale a postazioni remote. Quando si configura una flotta Anywhere, si aggiungono computer al parco computer registrandoli tramite Amazon GameLift Servers. Ogni computer è registrato con il relativo indirizzo IP (o nome DNS) in modo che Amazon GameLift Servers può stabilire una connessione con esso.

Puoi distribuire il software del server di gioco su una flotta Anywhere installandolo su ogni computer e avviando i processi del server di gioco. Ogni processo del server di gioco avviato stabilisce una connessione al Amazon GameLift Servers servizio e segnala la disponibilità a ospitare una sessione

di gioco. È possibile utilizzare gli strumenti di gestione e distribuzione della configurazione esistenti per gestire le attività di distribuzione iniziale e di gestione dell'host. Sono necessarie alcune attività aggiuntive da utilizzare con Amazon GameLift Servers, tra cui:

- Registra e annulla la registrazione dei computer per aggiungerli o rimuoverli dal parco dati.
- Conserva i token up-to-date di autenticazione su tutti i computer. I processi del server sul computer li utilizzano quando si connettono a Amazon GameLift Servers servizio.

> **Note**
>
> Opzionalmente, implementa la tua flotta Anywhere con Amazon GameLift Servers Agente per automatizzare queste attività di gestione chiave. Per informazioni, consulta Collabora con l'Amazon GameLift Serversagente.

Consulta questi argomenti su come configurare e gestire le flotte Anywhere:

- Roadmap di sviluppo per l'hosting con Anywhere Amazon GameLift Servers
- Roadmap di sviluppo per l'hosting ibrido con Amazon GameLift Servers
- Configurato per lo sviluppo iterativo con Amazon GameLift Servers Ovunque
- Crea un Amazon GameLift Servers Flotta ovunque
- Come funziona Amazon GameLift Servers la creazione della flotta
- Aggiornamento di una configurazione del Amazon GameLift Servers parco veicoli

Workflow di creazione di flotte ovunque

Flotte For Anywhere, Amazon GameLift Servers imposta solo la risorsa della flotta. L'utente configura e registra i computer con la flotta, installa il software del server di gioco e avvia i processi del server di gioco per ospitare sessioni di gioco.

1. Amazon GameLift Servers crea la risorsa della flotta nella regione di origine della flotta. Lo stato della flotta e lo stato della posizione personalizzata sono impostati su Nuovo.
2. Amazon GameLift Servers inizia a scrivere gli eventi nel registro degli eventi della flotta.
3. Dopo aver creato la risorsa del parco veicoli. Amazon GameLift Servers imposta lo stato della flotta su Attivo. A questo punto, puoi registrare nuovi computer con la flotta.

Crea un Amazon GameLift Servers Flotta ovunque

Questo argomento descrive come creare un Amazon GameLift Servers Flotta ovunque. Con una flotta Anywhere, puoi usare core Amazon GameLift Servers funzionalità di gestione delle sessioni di gioco mentre si ospitano sessioni di gioco con le proprie risorse di elaborazione. Crea una flotta Anywhere per il tuo hardware locale o altre risorse basate sul cloud.

Le flotte Anywhere vengono comunemente utilizzate insieme Amazon GameLift Servers flotte gestite in una soluzione di hosting ibrida. Forniscono inoltre ambienti di test utili per lo sviluppo di un gioco da ospitare con Amazon GameLift Servers. Consulta questi argomenti per saperne di più su quando e come incorporare Amazon GameLift Servers Anywhere flotte in una soluzione di hosting di giochi:

- Hosting ibrido
- Configurazione di una flotta di hosting con Amazon GameLift Servers
- Configurato per lo sviluppo iterativo con Amazon GameLift Servers Ovunque

Poiché le flotte Anywhere sono autogestite, la creazione di una flotta richiede del lavoro aggiuntivo. Per preparare una flotta Anywhere a ospitare sessioni di gioco e giocatori, devi completare le seguenti attività:

Argomenti

- Prima di iniziare
- Creare una posizione personalizzata
- Crea una flotta Anywhere
- Aggiungi un computer alla flotta
- Avvia un server di gioco

Prima di iniziare

Prima di creare una flotta Anywhere, esegui le seguenti operazioni. Per una guida più dettagliata, consulta Roadmap di sviluppo per l'hosting con Anywhere Amazon GameLift Servers oRoadmap di sviluppo per l'hosting ibrido con Amazon GameLift Servers.

- Integra il codice del tuo server di gioco con il Amazon GameLift Servers server SDK versione 5.x (o superiore). Non è necessario completare tutte le attività di integrazione del gioco, solo quelle necessarie per la creazione di un server di gioco. Una pratica comune consiste nel configurare

la macchina locale come una flotta Anywhere e utilizzare un'interfaccia a riga di comando per testare l'integrazione del server di gioco (vediConfigura i test locali con Amazon GameLift Servers Anywhere). È possibile incorporare componenti aggiuntivi (come Amazon GameLift Servers client di gioco abilitato) man mano che li sviluppi.

- Package del software del server di gioco per l'installazione sui computer della tua flotta Anywhere. Il pacchetto deve includere la versione integrata del server di gioco e tutto il software di supporto necessario per far funzionare il server di gioco.
- Decidi se utilizzare il Amazon GameLift Servers Agente della tua flotta Anywhere. L'Agent è uno strumento di gestione dei processi on-computing che automatizza alcune delle attività chiave relative alla gestione dei processi e dei calcoli del server da utilizzare con Amazon GameLift Servers. Per ulteriori informazioni, vedereCollabora con l'Amazon GameLift Serversagente.

Creare una posizione personalizzata

Crea una posizione personalizzata per rappresentare la posizione fisica delle tue risorse di elaborazione. Quando crei una flotta Anywhere, devi avere almeno una posizione personalizzata già definita. Puoi creare altre sedi personalizzate e aggiungerle a una flotta esistente in qualsiasi momento.

Per creare una sede personalizzata

Usa uno dei Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI) per creare una posizione personalizzata.

Console

Nella Amazon GameLift Servers console, utilizza il riquadro di navigazione per aprire la pagina Posizioni. Scegli Crea posizione per aprire la finestra di dialogo Crea.

1. Nella finestra di dialogo, inserite un nome per la località. Come procedura consigliata, utilizzate un nome che descriva una posizione significativa per un set di risorse di calcolo. Potrebbe trattarsi di posizioni geografiche, il nome di un data center o un altro identificatore di posizione. Amazon GameLift Servers aggiunge il nome della posizione personalizzata con custom -.
2. (Facoltativo) Aggiungi tag alla tua posizione personalizzata. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Assegna tag alle AWS risorse che desideri classificare in modi utili, ad esempio per scopo, proprietario o ambiente. Scegli Aggiungi nuovo tag per ogni tag che desideri aggiungere.

3. Scegli Create (Crea) .

AWS CLI

Crea una posizione personalizzata utilizzando il `create-location`comando. Fornisci un `location-name` valore, che deve iniziare con`custom-`. Come procedura consigliata, utilizzate un nome che descriva una posizione significativa per un set di risorse di elaborazione. Potrebbe trattarsi di posizioni geografiche, il nome di un data center o un altro identificatore di posizione.

```
aws gamelift create-location \
    --location-name custom-location-1
```

Output

```
{
    "Location": {
        "LocationName": "custom-location-1",
        "LocationArn": "arn:aws:gamelift:us-east-1:111122223333:location/custom-
location-1"
    }
}
```

Crea una flotta Anywhere

Crea una flotta Anywhere per un set di risorse di elaborazione di tua proprietà. Una nuova flotta Anywhere inizia vuota; aggiungi computer alla flotta registrandoli.

Al momento della creazione, una nuova flotta Anywhere passa rapidamente da uno stato all'altro della flotta. `NEW` `ACTIVE` Puoi aggiungere computer alla flotta una volta raggiunta la soglia. `ACTIVE`

Per creare una flotta Anywhere

Usa uno dei due Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI) per creare una flotta Anywhere.

Console

Nella Amazon GameLift Servers console, usa il pannello di navigazione per aprire la pagina Fleets. Scegli Crea flotta per avviare il flusso di lavoro di creazione della flotta.

Passaggio 1 Scegli il tipo di elaborazione

Seleziona l'opzione Anywhere e scegli Avanti.

Fase 2 Definire i dettagli della flotta

In questo passaggio, specifica alcune impostazioni chiave a livello di flotta.

1. Compila la sezione dei dettagli della flotta:
   a. Inserisci un nome per la flotta. Ti consigliamo di utilizzare uno schema di denominazione della flotta che faciliti l'identificazione dei tipi di flotta durante la visualizzazione degli elenchi di flotte.
   b. Fornisci una breve descrizione della flotta.
2. Imposta questi dettagli aggiuntivi opzionali in base alle esigenze. Puoi aggiornare queste impostazioni del parco veicoli in un secondo momento.
   a. Quando crei una flotta per la produzione o i test preliminari alla produzione, utilizza questa impostazione per specificare un valore di costo orario per i calcoli del parco macchine. Amazon GameLift Servers può utilizzare queste informazioni durante il processo di posizionamento della sessione di gioco per selezionare le risorse di hosting in base ai costi.
   b. Se desideri combinare i dati metrici per questo parco veicoli e altri, specifica il nome del gruppo Metric. Utilizza lo stesso nome di gruppo metrico per tutte le flotte che desideri combinare. Visualizza le metriche per il gruppo di metriche per vedere i dati aggregati.
3. Aggiungi tag opzionali alla tua posizione personalizzata. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Assegna tag alle AWS risorse che desideri classificare in modi utili, ad esempio per scopo, proprietario o ambiente. Scegli Aggiungi nuovo tag per ogni tag che desideri aggiungere.
4. Scegli Avanti per continuare il flusso di lavoro.

Passaggio 3 Seleziona posizioni personalizzate

In questo passaggio, identifica la posizione fisica dei computer che intendi aggiungere a questa flotta. Ora puoi specificare una o più sedi e aggiungere o rimuovere sedi in un secondo momento, se necessario.

1. In Posizioni personalizzate, seleziona una o più sedi per i computer della flotta. L'elenco include tutte le località personalizzate che sono state definite nella selezione Regione AWS corrente. Per definire una nuova posizione personalizzata da aggiungere al parco veicoli, scegli Crea sede.
2. Scegli Avanti per continuare il flusso di lavoro.

Passaggio 4: rivedi e crea

Controlla le impostazioni prima di creare la flotta.

Quando sei pronto per implementare la nuova flotta, scegli Crea. Amazon GameLift Servers avvia immediatamente il processo di attivazione della flotta, assegnando un ID univoco e impostando lo stato della flotta. `NEW` Puoi monitorare i progressi della flotta nella pagina Flotte.

AWS CLI

Usa il `create-fleet`comando per creare una flotta di tipi di calcolo. `ANYWHERE` Fornisci un nome e almeno una posizione personalizzata. Amazon GameLift Servers crea la risorsa Anywhere fleet con l'impostazione predefinita corrente Regione AWS (oppure puoi aggiungere un tag --region per specificarne un altro Regione AWS).

La seguente richiesta di esempio crea una nuova flotta con le impostazioni minime richieste. Sostituisci *`FleetName`* e *`custom-location`* con le tue informazioni.

```
aws gamelift create-fleet \
--name FleetName \
--compute-type ANYWHERE \
--locations "Location=custom-location"
```

Example response

```
{
    "FleetAttributes": {
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetArn": "arn:aws:gamelift:us-west-2:111122223333:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "Name": "HardwareAnywhere",
        "CreationTime": "2023-02-23T17:57:42.293000+00:00",
        "Status": "ACTIVE",
        "MetricGroups": [
```

```
            "default"
        ],
        "CertificateConfiguration": {
            "CertificateType": "DISABLED"
        },
        "ComputeType": "ANYWHERE"
    }
}
```

Al momento della creazione, una nuova flotta Anywhere passa rapidamente allo status di flotta`ACTIVE`. Puoi aggiungere computer alla flotta dopo che è stata `ACTIVE` raggiunta.

Nota che la risposta non include le ubicazioni del parco veicoli. Puoi recuperare i dettagli completi della flotta chiamando `describe-fleet-attributes`e `describe-fleet-location-attributes`.

Aggiungi un computer alla flotta

Per aggiungere una risorsa di calcolo a una flotta e prepararla per ospitare sessioni di gioco, esegui le seguenti attività:

- Registrate il computer con il parco macchine. La registrazione dice Amazon GameLift Servers quali risorse fisiche di hosting fanno parte della flotta.
- Richiedi un token di autenticazione per il calcolo. Ogni server di gioco in esecuzione sul computer necessita di questo token per connettersi al Amazon GameLift Servers servizio. I token di autenticazione sono temporanei e devono essere aggiornati regolarmente.

> **Note**
>
> Se stai distribuendo il software del tuo server di gioco con il Amazon GameLift Servers Agente, puoi saltare questo passaggio. L'agente registra automaticamente ogni calcolo e mantiene un token di autenticazione valido per il calcolo. Per informazioni, consulta Collabora con l'Amazon GameLift Serversagente.

Puoi registrare un calcolo e richiedere un token di autenticazione utilizzando la AWS CLI o effettuando chiamate programmatiche all'SDK per AWS Amazon GameLift Servers. Queste azioni non sono disponibili tramite Amazon GameLift Servers console.

Come procedura consigliata, consigliamo di automatizzare entrambe queste attività aggiungendo uno script di avvio a ciascun calcolo. Lo script di avvio chiama automaticamente entrambi i comandi `register-compute` e `get-compute-auth-token`. Puoi anche automatizzare le attività per aggiornare regolarmente il token di autenticazione per tutta la durata del calcolo e annullare la registrazione del calcolo allo spegnimento.

Ciascuna delle azioni di avvio restituisce valori specifici del calcolo che è necessario archiviare sul computer. Quando un processo del server di gioco viene avviato sul computer, deve passare questi valori come parametri del server durante l'inizializzazione di una connessione con Amazon GameLift Servers servizio (vedi ServerParameters nel riferimento all'SDK del server). Ti consigliamo di impostare questi valori specifici del calcolo (o le relative posizioni memorizzate) come variabili di ambiente. Se stai usando il Amazon GameLift Servers Agente, questo compito è gestito per te. I valori specifici del calcolo sono i seguenti:

- `register-compute`restituisce un valore per. `GameLiftServiceSdkEndpoint` Imposta questo valore sul parametro `webSocketUrl` del server.
- `compute-auth-token`restituisce il token di autenticazione. Imposta questo valore sul parametro `authToken` del server.

AWS CLI

Le seguenti istruzioni descrivono come inviare manualmente ogni richiesta utilizzando la AWS CLI.

Per registrare un calcolo

Chiama `register-compute`per registrare un computer. Identifica l'ID del parco macchine a cui aggiungere il calcolo. Fornisci le seguenti informazioni di calcolo: un nome, un indirizzo IP e una posizione significativi. La posizione del computer deve essere una posizione personalizzata già associata alla flotta. Se desideri utilizzare una posizione personalizzata diversa, utilizza il Amazon GameLift Servers console per aggiornare la flotta o chiamare il comando AWS CLI `create-fleet-locations`per aggiungere una posizione personalizzata alla flotta.

Nell'esempio seguente, sostituisci i valori segnaposto per il tuo calcolo e il tuo parco veicoli. Il `fleet-id` valore viene restituito quando si crea una flotta Anywhere. Puoi recuperare i dettagli completi della flotta chiamando `describe-fleet-attributes`e `describe-fleet-location-attributes`.

```
aws gamelift register-compute \
```

```
    --compute-name HardwareAnywhere \
    --fleet-id arn:aws:gamelift:us-east-1:111122223333:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \
    --ip-address 10.1.2.3 \
    --location custom-location-1
```

Output di esempio

```
{
    "Compute": {
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetArn": "arn:aws:gamelift:us-west-2:111122223333:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "ComputeName": "HardwareAnywhere",
        "ComputeArn": "arn:aws:gamelift:us-west-2:111122223333:compute/
HardwareAnywhere",
        "IpAddress": "10.1.2.3",
        "ComputeStatus": "Active",
        "Location": "custom-location-1",
        "CreationTime": "2023-02-23T18:09:26.727000+00:00",
        "GameLiftServiceSdkEndpoint": "wss://us-west-2.api.amazongamelift.com"
    }
}
```

Per richiedere un token di autenticazione

Chiama `get-compute-auth-token`per richiedere un token di autenticazione valido. Registra un calcolo. Identifica l'ID della flotta e il nome del calcolo.

Nell'esempio seguente, sostituisci i valori segnaposto per il calcolo e la flotta. Il `fleet-id` valore viene restituito quando si crea una flotta Anywhere. Puoi recuperare i dettagli completi della flotta `describe-fleet-attributes`chiamando. Per trovare informazioni di calcolo, chiama `list-compute`con l'ID della flotta per vedere tutti i computer registrati nel parco macchine.

```
aws gamelift get-compute-auth-token \
    --fleet-id arn:aws:gamelift:us-east-1:111122223333:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \
    --compute-name HardwareAnywhere
```

Output di esempio

```
{
```

```
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetArn": "arn:aws:gamelift:us-east-1:111122223333:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "ComputeName": "HardwareAnywhere",
    "ComputeArn": "arn:aws:gamelift:us-east-1:111122223333:compute/
HardwareAnywhere",
    "AuthToken": "0c728041-3e84-4aaa-b927-a0fb202684c0",
    "ExpirationTimestamp": "2023-02-23T18:47:54+00:00"
}
```

Avvia un server di gioco

Dopo aver creato una flotta Anywhere e aver aggiunto uno o più computer alla flotta, sei pronto per iniziare a utilizzare i tuoi server di gioco.

Passaggio 1 Installa il software del server di gioco

Installa la build del tuo server di gioco e tutto il software dipendente su ogni computer della tua flotta Anywhere. La build del server di gioco deve essere integrata con Amazon GameLift Servers server SDK versione 5.x (o successiva) con la funzionalità minima richiesta per comunicare con Amazon GameLift Servers servizio.

Passaggio 2 Prepara i tuoi computer per far funzionare un server di gioco

Assicurati che ogni computer sia registrato e disponga di un token di autenticazione valido. Se utilizzi degli script per gestire queste attività, assicurati che gli script vengano eseguiti su ogni computer prima di avviare qualsiasi processo del server di gioco.

Se hai distribuito il Amazon GameLift Servers Agent con il software del server di gioco, assicurati che l'eseguibile Agent venga avviato.

Passaggio 3 Avvia un processo del server di gioco

Esegui un'istanza del file eseguibile del tuo server di gioco su un computer. Se la build del server di gioco è integrata correttamente, il processo del server di gioco richiama l'azione SDK del server `InitSDK()` con una serie di parametri server validi. Quando il processo del server è pronto per ospitare una sessione di gioco, chiama`ProcessReady()`.

> **Note**
>
> Se hai distribuito il software del server di gioco con il Amazon GameLift Servers Agente, puoi saltare questo passaggio. L'agente avvia automaticamente i processi del server di gioco in base alle istruzioni di runtime fornite dall'utente.

È possibile monitorare l'avanzamento visualizzando le metriche dei processi del server per l'attivazione e i processi attivi del server. Per informazioni, consulta Parametri di Amazon GameLift Servers per parchi istanze. Se il processo del server di gioco non riesce a inizializzarsi, verifica che il processo stia recuperando i valori dei parametri del server corretti per il computer su cui è in esecuzione.

Astratto e Amazon GameLift Servers designazione della flotta con un alias

Un record Amazon GameLift Servers l'alias viene utilizzato per astrarre una destinazione di hosting. Le destinazioni di hosting raccontano Amazon GameLift Servers dove cercare le risorse disponibili per ospitare una nuova sessione di gioco per i giocatori. Gli alias sono utili nei seguenti scenari:

- Se il gioco non utilizza code con più flotte per il posizionamento delle sessioni di gioco, richiede nuove sessioni di gioco specificando un Amazon GameLift Servers ID della flotta. Durante la durata di un gioco, sostituirai la flotta più volte, per aggiornare una build del server, aggiornare l'hardware e il sistema operativo di hosting o risolvere problemi di prestazioni. Usa un alias per estrarre l'ID della flotta in modo da poter trasferire senza problemi il traffico dei giocatori da un parco giocatori esistente a quello nuovo.
- Se vuoi fare qualcosa di diverso, basta creare una nuova sessione di gioco quando un client di gioco ne richiede una. Ad esempio, potresti voler indirizzare i giocatori che utilizzano un out-of-date client verso un sito Web di aggiornamento.

Un alias deve specificare una strategia di routing. Esistono due tipi. Una semplice strategia di routing indirizza il traffico dei giocatori verso un ID flotta specifico, che puoi aggiornare per reindirizzare il traffico. Una strategia di routing del terminale invia un messaggio al client invece di creare una nuova sessione di gioco. Puoi modificare la strategia di routing di un alias in qualsiasi momento.

Se utilizzi una coda per il posizionamento delle sessioni di gioco, non hai bisogno di un alias per reindirizzare il traffico quando sostituisci una flotta. Con una coda, puoi semplicemente aggiungere la nuova flotta e rimuovere quella vecchia. Questa azione non è visibile ai giocatori, perché le nuove richieste di sessioni di gioco vengono soddisfatte automaticamente utilizzando la nuova flotta. Non ha alcun impatto sulle sessioni di gioco esistenti. Puoi identificare le destinazioni delle code utilizzando un ID flotta o un alias.

Argomenti

Crea un Amazon GameLift Servers alias

Questo argomento descrive come creare un Amazon GameLift Servers alias da utilizzare per il posizionamento delle sessioni di gioco.

Per creare un alias

Usa uno dei Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI) per creare un alias.

Console

Nella Amazon GameLift Servers console, usa il pannello di navigazione per aprire la pagina Alias.

1. Scegli Crea alias.
2. Inserisci un nome alias. Ti consigliamo di includere caratteristiche significative nel nome dell'alias per facilitare la visualizzazione di un elenco di alias.
3. Inserisci una descrizione dell'alias, se necessario.
4. Scegliete una strategia di routing per l'alias.
    a. Se scegli una strategia di routing semplice, seleziona un ID flotta dall'elenco da associare a questo alias. L'elenco include tutte le flotte attualmente selezionate. Regione AWSÈ necessario creare un alias nella stessa regione della flotta.
    b. Se scegli una strategia di routing del terminale, inserisci il valore di stringa che desideri Amazon GameLift Servers per tornare a un client di gioco in risposta a una richiesta di

sessione di gioco. Una richiesta con un alias di terminale genera un'eccezione con il messaggio incorporato.

5. (Facoltativo) Aggiungi tag alla risorsa alias. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Assegna tag alle AWS risorse che desideri classificare in modi utili, ad esempio per scopo, proprietario o ambiente. Scegli Aggiungi nuovo tag per ogni tag che desideri aggiungere.
6. Quando sei pronto per implementare la nuova flotta, scegli Crea.

AWS CLI

Usa il `create-alias`comando per creare un alias.Amazon GameLift Servers crea la risorsa alias con l'impostazione predefinita corrente Regione AWS (oppure puoi aggiungere un tag --region per specificarne uno diverso). Regione AWS

Come minimo, includi un nome alias e una strategia di routing. Per una strategia di routing semplice, specifica l'ID di una flotta nella stessa regione dell'alias. Per una strategia di routing dei terminali, fornisci una stringa di messaggi.

Modificare un alias

È possibile modificare un alias utilizzando Amazon GameLift Servers console o con il comando AWS CLI update-alias.

Questo argomento descrive come modificare un Amazon GameLift Servers alias da utilizzare per il posizionamento delle sessioni di gioco. Puoi apportare le seguenti modifiche:

Per modificare un alias

Usa uno dei Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI) per modificare un alias.

Console

Nella Amazon GameLift Servers console, usa il pannello di navigazione per aprire la pagina Alias.

1. Seleziona l'alias che desideri modificare e scegli Modifica. Se non vedi l'alias che desideri modificare, controlla quello attualmente selezionato. Regione AWS
2. Nella pagina Modifica alias, puoi apportare le seguenti modifiche:

- Cambia il nome dell'alias.
- Modificare la descrizione dell'alias.
- Cambia la strategia di routing da semplice a terminale o da terminale a semplice.
- Per un alias con una strategia di routing semplice, modifica l'ID della flotta a cui è associato l'alias.
- Per un alias con una strategia di routing dei terminali, modifica il testo del messaggio.

3. Scegli Save changes (Salva modifiche). Quando aggiorni l'ID della flotta per un alias con una semplice strategia di routing, potrebbero essere necessari fino a 2 minuti per completare la transizione. Durante questo periodo, potrebbero verificarsi nuovi posizionamenti nelle sessioni di gioco sulla vecchia flotta.

AWS CLI

Utilizzate il `update-alias`comando per apportare modifiche a una risorsa alias. È possibile aggiornare una risorsa alias con l'impostazione predefinita Regione AWS corrente oppure aggiungere un `--region` tag per specificarne uno diverso. Regione AWS

È possibile modificare le seguenti proprietà:

- Nome alias.
- Descrizione dell'alias.
- Tipo di strategia di routing. Assicurati di fornire un ID della flotta o una stringa di messaggi per la nuova strategia di routing.
- Fleet ID per una strategia di routing semplice esistente. L'ID della flotta deve trovarsi nella stessa regione dell'alias.
- Stringa di messaggio per una strategia di routing dei terminali esistente.

Gestione del posizionamento delle sessioni di gioco con Amazon GameLift Servers code

La coda delle sessioni di gioco è il meccanismo principale che Amazon GameLift Servers utilizza per cercare i server di gioco disponibili e sceglierli per ospitare nuove sessioni di gioco. Queues offre un modo molto più efficiente per elaborare un gran numero di richieste di sessioni di gioco e trovare posizioni per esse su più flotte di risorse di hosting. Se la tua soluzione di hosting utilizza più di una flotta e stai elaborando volumi elevati di richieste, probabilmente hai bisogno di una coda.

Quando il gioco desidera iniziare una nuova sessione di gioco per i giocatori, invia una richiesta di posizionamento al Amazon GameLift Servers servizio, che lo incanala in coda. La configurazione della coda determina quando e come vengono elaborate le richieste. Durante l'elaborazione di una richiesta di collocamento, Amazon GameLift Servers cerca in un insieme di flotte un server di gioco per ospitare la sessione di gioco. Il posizionamento ha esito positivo quando Amazon GameLift Servers trova un server di gioco disponibile e gli chiede di avviare una sessione di gioco.

Argomenti

Caratteristiche della coda

Un record Amazon GameLift Servers la coda delle sessioni di gioco è una risorsa AWS cloud. Puoi creare una coda in qualsiasi momento Regione AWS Amazon GameLift Servers supporti (vediAmazon GameLift Serverssedi di assistenza). Le richieste di posizionamento delle sessioni di gioco vengono inviate a quella posizione ed elaborate lì.

L'automazione del posizionamento delle sessioni di gioco con le code offre vantaggi significativi sia per gli sviluppatori di giochi che per i giocatori. Ciò include:

- Le code offrono il «miglior posizionamento possibile». Durante l'elaborazione delle richieste di posizionamento delle sessioni di gioco, una coda utilizza il Amazon GameLift Servers Algoritmo

FleetIQ per dare priorità ai posizionamenti in base a una serie di preferenze definite, tra cui costo, posizione e latenza del giocatore.

- Le code supportano le flotte Spot per contribuire a ridurre i costi di hosting dei giochi. Puoi configurare le code con le flotte AWS Spot, che spesso offrono costi di hosting notevolmente inferiori, e con le flotte On-Demand. Poiché il basso costo è uno dei criteri chiave per i posizionamenti, le code possono sempre sfruttare le differenze di costo.
- Le code possono inserire nuovi giochi più velocemente in caso di forte richiesta. Configurando una coda con più flotte, offri opzioni più flessibili per il posizionamento delle sessioni di gioco. Ma le flotte aggiuntive forniscono anche la capacità di backup necessaria quando la domanda aumenta. Per qualsiasi richiesta di collocamento, se Amazon GameLift Servers non è possibile collocare una sessione di gioco nella posizione preferita, passa automaticamente alla valutazione di altre località.
- Le code possono rendere più resistente la disponibilità dei server di gioco. Le interruzioni possono verificarsi. Con una coda composta da più flotte, un rallentamento o un'interruzione non devono influire sull'accesso dei giocatori al gioco. Configurando la coda con flotte capienti in diverse Regioni AWS zone di disponibilità, puoi fare in modo che i giocatori possano sempre trovare una sessione di gioco a cui partecipare.
- Ottieni metriche sui posizionamenti delle sessioni di gioco e sulle prestazioni in coda. Amazon GameLift Servers emette metriche sulla coda, tra cui statistiche sui piazzamenti riusciti e non riusciti, il numero di richieste in coda e il tempo medio che le richieste trascorrono in coda. È possibile visualizzare queste metriche nella Amazon GameLift Servers console o in CloudWatch.

Per iniziare a creare una coda iniziale di base, consulta. Crea una coda per le sessioni di gioco

Crea una coda per le sessioni di gioco

Le code vengono utilizzate per effettuare nuove sessioni di gioco su più flotte e località. Il gioco avvia nuove sessioni di gioco inviando richieste di posizionamento a una coda. Una coda è configurata con istruzioni su come elaborare le richieste. Scopri di più sull'avvio delle richieste di posizionamento per le sessioni di gioco in. Crea sessioni di gioco

Per creare una coda per le sessioni di gioco

Queste istruzioni illustrano come creare una coda di lavoro semplice con impostazioni di configurazione minime e impostazioni predefinite. Esistono diverse opzioni per personalizzare la configurazione di una coda. Queste opzioni ti aiutano a ottenere i migliori posizionamenti possibili in base alle esigenze del tuo gioco. Per ulteriori informazioni sulla personalizzazione delle code per il

gioco, consulta. Personalizza una coda di sessioni di gioco Puoi aggiornare la maggior parte delle impostazioni di configurazione delle code in qualsiasi momento.

Puoi creare una coda di sessioni di gioco utilizzando la Amazon GameLift Servers console o la AWS CLI.

Console

Nella Amazon GameLift Serversconsole, seleziona una AWS regione in cui lavorare. Apri la barra di navigazione a sinistra della console e scegli Code.

1. Nella pagina Code, scegli Crea coda per avviare il flusso di lavoro.
2. In Impostazioni coda, inserisci le seguenti impostazioni:
    a. Immettete un nome per la coda. Questo nome deve essere univoco rispetto a Regione AWS quello in cui stai creando la coda.
    b. Mantieni l'impostazione di timeout predefinita, che è 600 secondi (o 10 minuti). Questo valore controlla per quanto tempo si Amazon GameLift Servers cerca di effettuare una nuova sessione di gioco prima di interromperla. Amazon GameLift Serverscerca le risorse disponibili fino al timeout della richiesta. È possibile aggiornare l'impostazione del timeout di una coda in qualsiasi momento.
    c. Salta la sezione relativa ai criteri di latenza di Player. Una coda utilizza i criteri di latenza solo quando riceve richieste di posizionamento che includono dati sulla latenza dei giocatori. Puoi aggiungere politiche di latenza a una coda in qualsiasi momento. Per ulteriori informazioni sulla creazione di criteri di latenza, consulta. Crea una politica di latenza dei giocatori
3. Salta la sezione Posizionamento delle sessioni di gioco per utilizzare l'impostazione predefinita di Tutte le posizioni. Questa impostazione consente di creare un elenco di posizioni consentite in cui la coda può effettuare posizionamenti (chiamata anche configurazione dei filtri). Per ulteriori informazioni sulla definizione delle priorità in base alla posizione e alle configurazioni dei filtri, consulta. Assegna priorità ai posizionamenti in base alla posizione
4. In Ordine di destinazione, aggiungi una o più flotte alla coda. Puoi identificare le flotte utilizzando fleet IDs o ARNs, o utilizzando un alias di flotta. Quando aggiungi più flotte, tieni presente che tutte dovrebbero eseguire build di gioco simili ed essere compatibili con qualsiasi client di gioco che utilizza questa coda. Inoltre, tutte le flotte in coda devono avere la stessa configurazione di certificati.

a. Seleziona la regione in cui è stata creata la flotta o l'alias. Per una flotta con più sedi, questa è la regione «di origine».

b. Per Tipo di destinazione, seleziona una flotta o un alias.

c. Le selezioni relative alla regione e al tipo compongono un elenco a discesa di flotte o alias esistenti. Selezionane uno da designare come destinazione della coda.

d. Per specificare un'altra flotta o un alias per la coda, scegli Aggiungi destinazione e ripeti i passaggi precedenti.

e. Dopo aver aggiunto un elenco di destinazioni, utilizza la drag-and-drop funzione per riordinare le destinazioni. Amazon GameLift Serversutilizza questo ordine per assegnare priorità ai posizionamenti in base alla destinazione.

5. Salta la sezione Priorità di posizionamento della sessione di gioco per mantenere l'ordine di priorità predefinito. Questa impostazione ti consente di personalizzare il modo in cui Amazon GameLift Servers scegli dove cercare le risorse di hosting disponibili per i nuovi posizionamenti delle sessioni di gioco. Per ulteriori informazioni sull'assegnazione di priorità ai posizionamenti, consulta. Dai priorità al posizionamento delle sessioni di gioco Puoi aggiornare le priorità di posizionamento di una coda in qualsiasi momento.

6. In Ordine di ubicazione, mantieni i valori predefiniti. Questa impostazione viene utilizzata per stabilire le priorità in base alla posizione del parco veicoli. Fornisce l'ordine di posizione da utilizzare. Quando si utilizzano le impostazioni di priorità predefinite, la posizione viene utilizzata come punto di partenza quando la destinazione preferita è una flotta con più sedi.

7. Salta la sezione opzionale delle impostazioni di notifica degli eventi. Le notifiche degli eventi sono necessarie per le code che elaborano un volume elevato di richieste di posizionamento. Per le code che elaborano volumi ridotti, ad esempio per scopi di sviluppo o test, puoi monitorare lo stato delle richieste di collocamento effettuando un sondaggio con. DescribeGameSessionPlacement Per ulteriori dettagli, consulta Imposta la notifica degli eventi per il posizionamento della sessione di gioco. Puoi aggiornare le impostazioni di notifica degli eventi di una coda in qualsiasi momento.

8. Scegli Crea per generare una nuova coda con una personalizzazione minima.

AWS CLI

Example Crea una coda

L'esempio seguente crea una coda di sessioni di gioco con queste configurazioni:

- Un timeout di cinque minuti.
- Due destinazioni per la flotta.
- Filtra per consentire solo i posizionamenti in queste località:`us-east-1`,`us-east-2`. `us-west-2`, e. `ca-central-1`
- Ordine di priorità basato sul costo e quindi sulle ubicazioni in un ordine specificato.

```
aws gamelift create-game-session-queue \
    --name "sample-test-queue" \
    --timeout-in-seconds 300 \
    --destinations DestinationArn="arn:aws:gamelift:us-east-1:111122223333:fleet/
fleet-772266ba-8c82-4a6e-b620-a74a62a93ff8" DestinationArn="arn:aws:gamelift:us-
east-1:111122223333:fleet/fleet-33f28fb6-aa8b-4867-85b4-ceb217bf5994" \
    --filter-configuration "AllowedLocations=us-east-1, ca-central-1, us-east-2, us-
west-2" \
    --priority-configuration PriorityOrder="COST","LOCATION",LocationOrder="us-
east-1","us-east-2","ca-central-1","us-west-2" \
    --notification-target "arn:aws:sns:us-east-1:111122223333:gamelift-test.fifo"
```

> **Note**
>
> Puoi ottenere i valori ARN di fleet e alias chiamando uno dei describe-fleet-attributesdue o describe-alias con l'ID fleet o alias.

Se la `create-game-session-queue` richiesta ha esito positivo, Amazon GameLift Servers restituisce un GameSessionQueueoggetto con la nuova configurazione della coda. È ora possibile inviare richieste alla coda utilizzando. StartGameSessionPlacement

Example Crea una coda con le politiche di latenza dei giocatori

L'esempio seguente crea una coda di sessioni di gioco con queste configurazioni:

- Un timeout di dieci minuti
- Tre destinazioni per la flotta
- Una serie di politiche di latenza dei giocatori

```
aws gamelift create-game-session-queue \
```

```
    --name "matchmaker-queue" \
    --timeout-in-seconds 600 \
    --destinations DestinationArn=arn:aws:gamelift:us-east-1::alias/alias-a1234567-
b8c9-0d1e-2fa3-b45c6d7e8910 \
                DestinationArn=arn:aws:gamelift:us-west-2::alias/alias-b0234567-
c8d9-0e1f-2ab3-c45d6e7f8901 \
                DestinationArn=arn:aws:gamelift:us-west-2::fleet/fleet-f1234567-
b8c9-0d1e-2fa3-b45c6d7e8912 \
    --player-latency-policies
 "MaximumIndividualPlayerLatencyMilliseconds=50,PolicyDurationSeconds=120" \

 "MaximumIndividualPlayerLatencyMilliseconds=100,PolicyDurationSeconds=120" \
                "MaximumIndividualPlayerLatencyMilliseconds=150" \
```

Se la `create-game-session-queue` richiesta ha esito positivo, Amazon GameLift Servers restituisce un GameSessionQueueoggetto con la nuova configurazione della coda.

Personalizza una coda di sessioni di gioco

Questo argomento descrive come personalizzare le code delle sessioni di gioco per prendere le migliori decisioni possibili sul posizionamento delle sessioni di gioco. Per ulteriori informazioni sulle code delle sessioni di gioco e sul loro funzionamento, consulta. Gestione del posizionamento delle sessioni di gioco con Amazon GameLift Servers code

Queste Amazon GameLift Servers funzionalità richiedono code:

- Abbinamento con FlexMatch
- Progettare una coda per le istanze Spot

Argomenti

- Procedure consigliate per le code delle sessioni di Amazon GameLift Servers gioco
- Definisci l'ambito di una coda
- Crea una coda con più sedi
- Dai priorità al posizionamento delle sessioni di gioco
- Valuta le metriche della coda
- Progettare una coda per le istanze Spot

Procedure consigliate per le code delle sessioni di Amazon GameLift Servers gioco

Una coda per le sessioni di gioco contiene un elenco di flotte in cui è Amazon GameLift Servers possibile inserire nuove sessioni di gioco. Ogni flotta può avere risorse di hosting distribuite in più località geografiche. Quando si sceglie una posizione, la coda seleziona una flotta e un'ubicazione della flotta in base a una serie di priorità impostate per la flotta.

Prendi in considerazione le seguenti linee guida e best practice:

- Aggiungi flotte in località adatte ai tuoi giocatori. Puoi aggiungere flotte e alias in qualsiasi località disponibile. La posizione è importante se effettui posizionamenti in base alla latenza riportata dai giocatori.
- Usa gli alias per tutte le flotte. Assegna un alias a ogni flotta in coda e usa i nomi degli alias per impostare le destinazioni nella coda.
- Usa una build o uno script di gioco uguali o simili per tutte le flotte. La coda potrebbe mettere i giocatori in sessioni di gioco su qualsiasi flotta in coda. I giocatori devono essere in grado di giocare in qualsiasi sessione di gioco su qualsiasi flotta.
- Crea flotte in almeno due località. Avendo server di gioco ospitati in almeno un'altra località, riduci l'impatto delle interruzioni regionali sui tuoi giocatori. È possibile ridurre le dimensioni delle flotte di backup e utilizzare la scalabilità automatica per aumentare la capacità in caso di aumento dell'utilizzo.
- Dai priorità al posizionamento della sessione di gioco. Una coda dà priorità alle scelte di posizionamento in base a diversi elementi, incluso l'ordine dell'elenco delle destinazioni.
- Crea la tua coda nella stessa posizione del servizio clienti. Mettendo la coda in una posizione vicina al servizio clienti, puoi ridurre al minimo la latenza di comunicazione.
- Utilizza flotte con più sedi. Usa la configurazione del filtro di coda per evitare che la coda collochi le sessioni di gioco in posizioni specifiche. Puoi utilizzare almeno due flotte con più sedi con sedi diverse per mitigare l'impatto dei piazzamenti di gioco durante un'interruzione regionale.
- Utilizza la stessa impostazione del certificato TLS per tutte le flotte. I client di gioco che si connettono alle sessioni di gioco delle tue flotte devono disporre di protocolli di comunicazione compatibili.

Definisci l'ambito di una coda

La popolazione di giocatori del tuo gioco potrebbe avere gruppi di giocatori che non dovrebbero giocare insieme. Ad esempio, se pubblichi il gioco in due lingue, ogni lingua dovrebbe avere i propri server di gioco.

Per impostare il posizionamento delle sessioni di gioco in base al numero di giocatori, crea una coda separata per ogni segmento di giocatori. Analizza ogni coda per posizionare i giocatori nei server di gioco corretti. Alcuni metodi comuni per definire l'ambito delle code includono:

- Per località geografiche. Quando distribuisci i tuoi server di gioco in più aree geografiche, potresti creare code per i giocatori in ogni località per ridurre la latenza dei giocatori.
- Tramite build o varianti dello script. Se hai più di una variante del tuo server di gioco, potresti supportare gruppi di giocatori che non possono giocare nelle stesse sessioni di gioco. Ad esempio, le build o gli script dei server di gioco potrebbero supportare lingue o tipi di dispositivi diversi.
- Per tipi di eventi. Puoi creare una coda speciale per gestire le partite per i partecipanti a tornei o altri eventi speciali.

Progetta code multiple

A seconda del gioco e dei giocatori, potresti voler creare più di una coda per le sessioni di gioco. Quando il servizio client di gioco richiede una nuova sessione di gioco, specifica quale coda di sessione di gioco utilizzare. Per aiutarti a determinare se utilizzare più code, considera:

- Varianti del tuo server di gioco. Puoi creare una coda separata per ogni variante del tuo server di gioco. Tutte le flotte in coda devono utilizzare server di gioco compatibili. Questo perché i giocatori che utilizzano la coda per partecipare alle partite devono poter giocare su qualsiasi server di gioco della coda.
- Gruppi di giocatori diversi. Puoi personalizzare Amazon GameLift Servers il posizionamento delle sessioni di gioco in base al gruppo di giocatori. Ad esempio, potresti aver bisogno di code personalizzate per determinate modalità di gioco che richiedono un tipo di istanza o una configurazione di runtime speciali. Oppure, potresti aver bisogno di una coda speciale per gestire i piazzamenti per un torneo o un altro evento.
- Metriche della coda delle sessioni di gioco. Puoi impostare le code in base a come desideri raccogliere le metriche di posizionamento delle sessioni di gioco. Per ulteriori informazioni, consulta Parametri Amazon GameLift Servers per code.

Crea una coda con più sedi

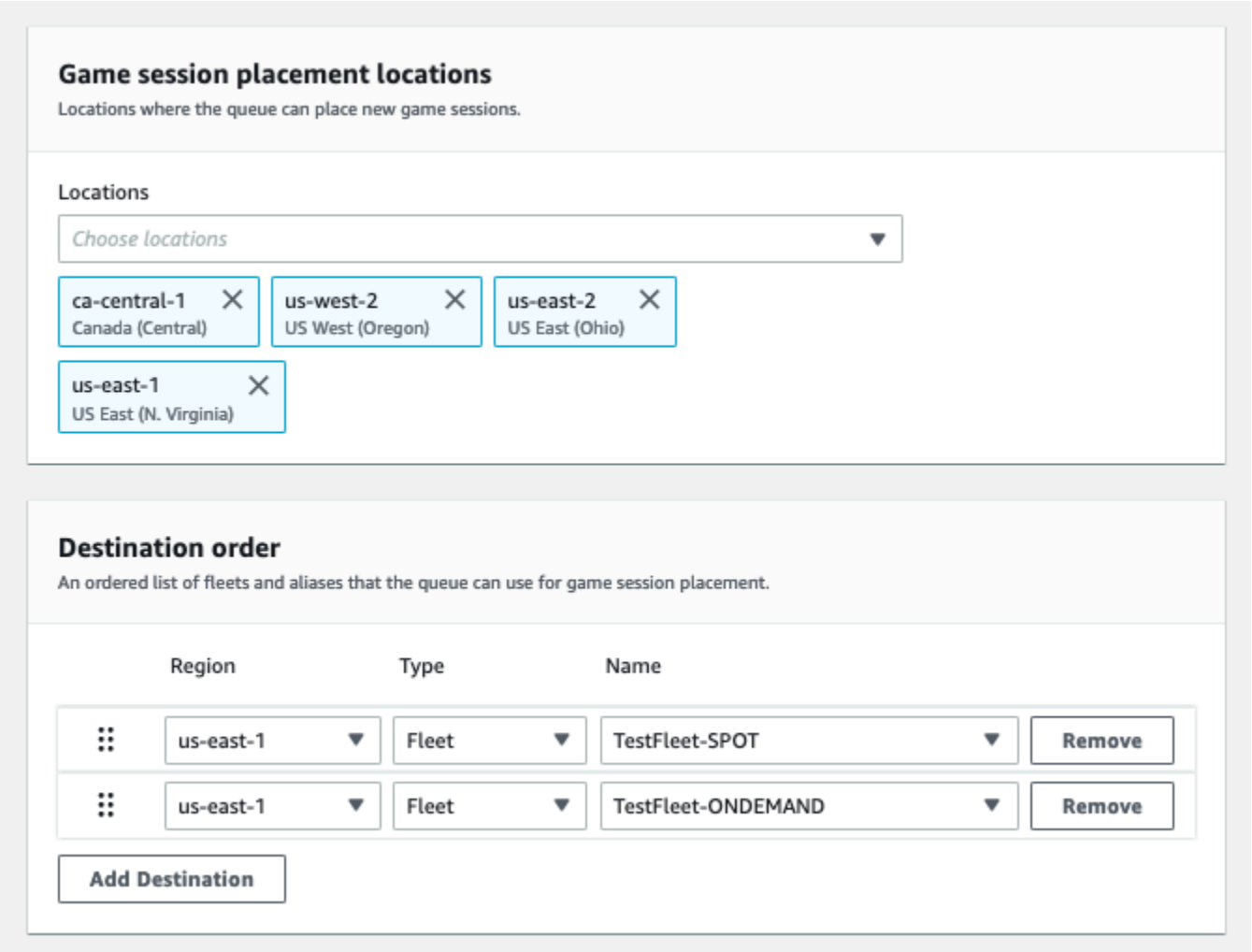
Consigliamo un design con più sedi per tutte le code. Questo design può migliorare la velocità di posizionamento e la resilienza dell'hosting. È necessario un design multi-location per utilizzare i dati di latenza dei giocatori per coinvolgere i giocatori in sessioni di gioco con una latenza minima. Se stai creando code con più sedi che utilizzano flotte di istanze Spot, segui le istruzioni riportate in. Tutorial: creare una Amazon GameLift Servers coda con le istanze Spot

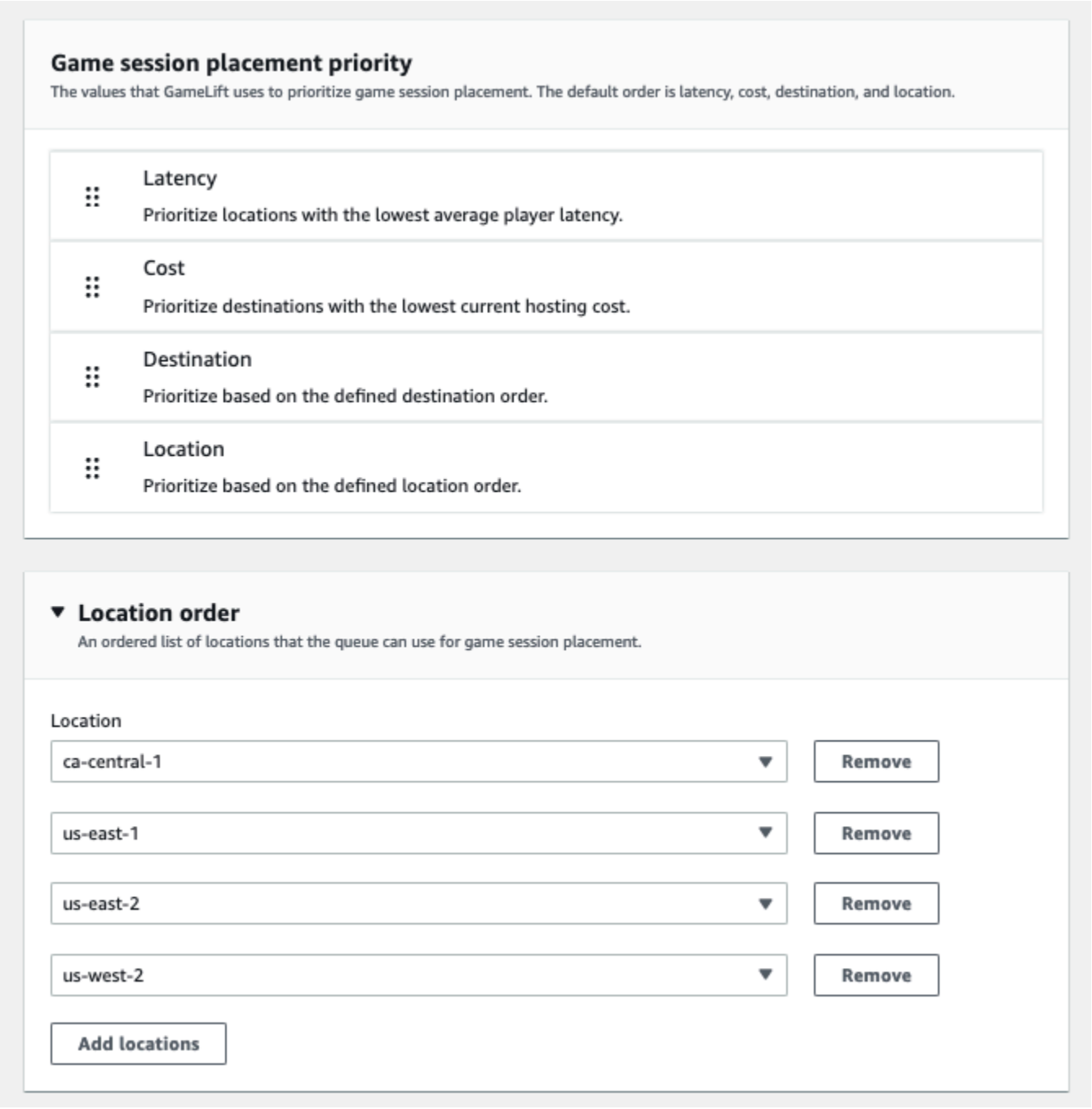
Un modo per creare una coda con più sedi consiste nell'aggiungere un parco veicoli con più sedi a una coda. In questo modo, la coda può collocare sessioni di gioco in qualsiasi postazione della flotta. Puoi anche aggiungere altre flotte con configurazioni diverse o ubicazioni principali per ridondarle. Se utilizzi una flotta di istanze Spot con più sedi, segui le best practice e includi una flotta di istanze On-Demand con le stesse sedi.

L'esempio seguente illustra il processo di progettazione di una coda di base con più sedi. In questo esempio, utilizziamo due flotte: una flotta di istanze Spot e una flotta di istanze On-Demand. Ogni flotta ha le seguenti località Regioni AWS di collocamento:`us-east-1`, `us-east-2ca-central-1`, e. `us-west-2`

Per creare una coda di base con più sedi con flotte con più sedi

1. Scegli una posizione in cui creare la coda. È possibile ridurre al minimo la latenza delle richieste posizionando la coda in una posizione vicina a dove è stato distribuito il servizio client. In questo esempio, creiamo la coda in. `us-east-1`
2. Crea una nuova coda e aggiungi le tue flotte con più sedi come destinazioni di coda. L'ordine di destinazione determina come Amazon GameLift Servers si svolgono le sessioni di gioco. In questo esempio, elenchiamo prima il parco istanze Spot e poi il parco istanze On-Demand.
3. Definisci l'ordine di priorità di posizionamento delle sessioni di gioco della coda. Questo ordine determina dove la coda cerca per primo un server di gioco disponibile. In questo esempio, utilizziamo l'ordine di priorità predefinito.
4. Definire l'ordine delle ubicazioni. Se non definisci l'ordine delle località, Amazon GameLift Servers utilizza le posizioni in ordine alfabetico.


Game session placement locations
Locations where the queue can place new game sessions.
Locations
Choose locations
ca-central-1
Canada (Central)
us-west-2
US West (Oregon)
us-east-2
US East (Ohio)
us-east-1
US East (N. Virginia)
Destination order
An ordered list of fleets and aliases that the queue can use for game session placement.
Region
Type
Name
us-east-1
Fleet
TestFleet-SPOT
Remove
us-east-1
Fleet
TestFleet-ONDEMAND
Remove
Add Destination



Dai priorità al posizionamento delle sessioni di gioco

Amazon GameLift Serversutilizza un algoritmo per determinare come dare priorità alle destinazioni di una coda e determinare dove collocare una nuova sessione di gioco. L'algoritmo si basa su un insieme ordinato di criteri. È possibile utilizzare l'ordine di priorità predefinito oppure personalizzare l'ordine. È possibile modificare l'ordine di priorità di una coda in qualsiasi momento.

Ordine di priorità predefinito

1. Latenza: se la richiesta di posizionamento della sessione di gioco include dati di latenza specifici per località per i giocatori, Amazon GameLift Servers calcola la latenza media dei giocatori in ciascuna località e tenta di collocare una sessione di gioco in una località della flotta con la media più bassa.
2. Costo: se una richiesta non include dati sulla latenza o se più flotte hanno la stessa latenza, valuta il costo di hosting di ciascuna flotta. Amazon GameLift Servers Il costo di hosting di una flotta varia in base al tipo di flotta (Spot o On-Demand), al tipo di istanza e alla posizione.
3. Destinazione: se più flotte hanno latenza e costi uguali, assegna la Amazon GameLift Servers priorità alle flotte in base all'ordine di destinazione indicato nella configurazione della coda.
4. Ubicazione: per le code con flotte con più sedi, se tutti gli altri criteri sono uguali, assegna la Amazon GameLift Servers priorità alle posizioni della flotta in base all'ordine alfabetico.

Personalizza il modo in cui una coda assegna la priorità ai posizionamenti delle sessioni di gioco

Puoi scegliere di personalizzare il modo in cui una coda assegna la priorità ai criteri di posizionamento. La coda applica la prioritizzazione personalizzata a tutte le richieste di posizionamento delle sessioni di gioco che riceve.

> **Note**
>
> Se crei una configurazione di priorità personalizzata e non includi tutti e quattro i criteri, aggiunge Amazon GameLift Servers automaticamente i criteri mancanti nell'ordine predefinito.

Per personalizzare la configurazione della priorità di una coda

Usa la Amazon GameLift Serversconsole o AWS Command Line Interface (AWS CLI) per creare una configurazione di priorità personalizzata.

Console

Nella Amazon GameLift Serversconsole, è possibile personalizzare le priorità di una coda quando si crea una nuova coda o si aggiorna una coda esistente. Seleziona una AWS regione in cui lavorare.

Apri la barra di navigazione a sinistra della console e scegli Code. Nella pagina Code, seleziona una coda esistente e scegli Modifica.

1. Vai alla sezione Priorità del posizionamento della sessione di gioco. Trascina e rilascia ogni criterio di priorità per creare l'ordine desiderato.
2. Vai alla sezione Ordine di ubicazione. Aggiungi tutte le sedi a cui desideri dare la priorità. Questo elenco è utile quando la coda ha flotte con più sedi. È necessario specificare almeno una posizione. Alle posizioni specificate qui viene data priorità per prime, seguite da tutte le altre posizioni nelle destinazioni della coda.
3. Scegli Save changes (Salva modifiche).

AWS CLI

Utilizzate il update-game-session-queuecomando con l'`--priority-configuration`opzione per personalizzare l'ordine di priorità di una coda. Amazon GameLift Serversaggiorna una coda nella AWS regione predefinita corrente oppure puoi aggiungere un `--region` tag per specificare una regione diversa AWS .

La seguente richiesta di esempio aggiunge o aggiorna la configurazione di priorità per una coda specificata

```
aws gamelift update-game-session-queue \
    --name "example-queue-with-priority"
    --priority-configuration
 PriorityOrder="COST','LOCATION","DESTINATION",LocationOrder="us-east-1","us-
east-2","ca-central-1","us-west-2" \
```

Dai priorità ai posizionamenti in base alla latenza del giocatore

Se vuoi offrire ai tuoi giocatori la migliore esperienza di gioco possibile e garantire una latenza minima, procedi nel seguente modo durante la configurazione del sistema di posizionamento delle sessioni di gioco:

- Imposta la coda per dare priorità alla latenza quando scegli dove collocare le sessioni di gioco. Per impostazione predefinita, la latenza è in cima all'elenco delle priorità. Puoi anche personalizzare la configurazione delle priorità della coda e scegliere dove mettere la latenza in ordine di priorità.
- Imposta le politiche di latenza dei giocatori per la tua coda. Le politiche di latenza ti consentono di impostare limiti rigidi sulla quantità di latenza da consentire nel posizionamento di una sessione di

gioco. Se non Amazon GameLift Servers riesci a completare una sessione di gioco senza superare i limiti, la richiesta di posizionamento scadrà e avrà esito negativo. Puoi impostare un'unica politica di latenza oppure puoi creare una serie di politiche che riducono gradualmente il limite di latenza nel tempo. Con una serie di policy, puoi specificare limiti di latenza iniziale molto bassi e comunque soddisfare i giocatori con latenze più elevate dopo un breve periodo di tempo. Per i dettagli sulla creazione di politiche di latenza, consulta. Crea una politica di latenza dei giocatori

- Quando effettui richieste di posizionamento delle sessioni di gioco (vedi StartGameSessionPlacement), includi i dati sulla latenza per ogni giocatore. I dati sulla latenza dei giocatori includono un valore per ogni possibile posizione in cui potrebbe essere tenuta una sessione di gioco. Ad esempio, per una coda che colloca le sessioni di gioco in Regioni AWS us-east-2 e ca-central-1, i dati di latenza potrebbero essere simili ai seguenti:

```
"PlayerLatencies": [
    { "LatencyInMilliseconds": 100, "PlayerId": "player1", "RegionIdentifier": "us-
east-2" },
    { "LatencyInMilliseconds": 100, "PlayerId": "player1", "RegionIdentifier": "ca-
central-1" },
    { "LatencyInMilliseconds": 150, "PlayerId": "player2", "RegionIdentifier": "us-
east-2" },
    { "LatencyInMilliseconds": 150, "PlayerId": "player2", "RegionIdentifier": "ca-
central-1" }
  ]
```

  Per ottenere misurazioni accurate della latenza, utilizzate i beacon ping UDP. Amazon GameLift Servers Questi endpoint consentono di misurare l'effettiva latenza della rete UDP tra i dispositivi dei giocatori e ciascuna delle potenziali sedi di hosting, con conseguenti decisioni di posizionamento più accurate rispetto all'utilizzo dei ping ICMP. Per ulteriori informazioni sull'utilizzo dei beacon ping UDP per misurare la latenza, fare riferimento a. Fari di ping UDP

Assegna priorità ai posizionamenti in base alla posizione

Puoi configurare una coda per posizionare le sessioni di gioco in base a un elenco di posizioni geografiche con priorità. La posizione è uno dei criteri che determinano il modo in cui una coda sceglie dove collocare una nuova sessione di gioco. Per impostazione predefinita, la posizione ha la priorità al quarto posto, dopo latenza, costo e destinazione.

Per quanto riguarda il posizionamento della sessione di gioco, destinazione e luogo hanno significati leggermente diversi:

- La destinazione si riferisce a una flotta specifica e include tutte le risorse di hosting della flotta, ovunque siano dispiegate. Quando stabilisci le priorità in base alla destinazione, Amazon GameLift Servers potresti effettuare un posizionamento in qualsiasi località della flotta. Le flotte gestite da più sedi e le flotte Anywhere possono disporre di risorse di hosting distribuite in una o più sedi.
- La posizione si riferisce a una posizione geografica specifica in cui vengono distribuite le risorse di hosting di una flotta. Una flotta può avere più sedi Regioni AWS, tra cui Local Zones o sedi personalizzate (per una flotta Anywhere). Una flotta gestita da un'unica sede ha un'unica sede ed è sempre una Regione AWS. Una flotta gestita da più sedi ha una regione di origine e può avere sedi remote. Una flotta Anywhere ha una o più sedi personalizzate.

Quando assegna la priorità ai posizionamenti in base alla località, Amazon GameLift Servers cerca tutte le destinazioni in coda che includono la posizione prioritaria e cerca in esse una risorsa di hosting disponibile. Se ci sono più destinazioni con la posizione prioritaria, Amazon GameLift Servers passa ai criteri di priorità successivi (costo, latenza, destinazione).

Esistono diversi modi per influenzare il modo in cui viene assegnata la priorità alle posizioni di una coda

- Configura il modo in cui la coda gestisce tutte le richieste di posizionamento delle sessioni di gioco:
  - Aggiungi una configurazione prioritaria alla coda. La configurazione prioritaria di una coda include un elenco ordinato di posizioni. È possibile specificare una o più posizioni a cui dare la priorità. Questo elenco non esclude alcuna località, indica semplicemente Amazon GameLift Servers dove cercare prima una risorsa di hosting disponibile. Un uso comune di un elenco di sedi ordinato è quando si desidera incanalare la maggior parte del traffico verso una o più località geografiche specifiche e utilizzare posizioni aggiuntive come capacità di backup. Aggiungi una configurazione prioritaria UpdateGameSessionQueuechiamando.
  - Aggiungi una configurazione di filtro alla coda. Una configurazione di filtro è un elenco di elementi consentiti per la coda. Indica Amazon GameLift Servers di ignorare tutte le posizioni che non sono nell'elenco quando si cerca una risorsa di hosting disponibile. Esistono due usi comuni per la configurazione di un filtro. Innanzitutto, per le flotte con più sedi, potresti utilizzare un filtro per escludere alcune sedi della flotta. In secondo luogo, potresti voler impedire temporaneamente i posizionamenti in una determinata località; ad esempio, una località potrebbe presentare problemi transitori. Poiché è possibile aggiornare la configurazione dei filtri di una coda in qualsiasi momento, è possibile aggiungere e rimuovere facilmente le posizioni in base alle esigenze. Aggiungi una configurazione di filtro UpdateGameSessionQueuechiamando.

- Utilizza istruzioni speciali per le richieste di collocamento individuali:
  - Includi un elenco di priorità prioritarie in una richiesta di posizionamento per una sessione di gioco. Puoi fornire un elenco alternativo di sedi con priorità per qualsiasi StartGameSessionPlacementrichiesta. Questo elenco sostituisce efficacemente la prioritizzazione configurata della coda per le posizioni solo per quella richiesta. Non ha alcun impatto su altre richieste. Questa funzionalità di override ha alcuni requisiti:
    - Usa un elenco di override solo con una coda che ha una configurazione di priorità impostata `LOCATION` come prima priorità.
    - Non includere i dati sulla latenza dei giocatori nella stessa richiesta di posizionamento. L'inclusione dei dati sulla latenza crea conflitti quando si assegnano priorità a posizioni che Amazon GameLift Servers non possono essere risolte.
    - Decidi come procedere se non riesce Amazon GameLift Servers a trovare una risorsa disponibile nell'elenco delle eccezioni di priorità. Scegli se tornare alle altre posizioni della coda o limitare i posizionamenti all'elenco delle eccezioni. Per impostazione predefinita, Amazon GameLift Servers torna indietro per tentare di posizionarlo nelle altre posizioni della coda.
    - Aggiorna la configurazione dei filtri della coda in base alle esigenze, ad esempio aggiungendo posizioni nell'elenco delle sostituzioni. L'elenco delle eccezioni non invalida l'elenco dei filtri.

Crea una politica di latenza dei giocatori

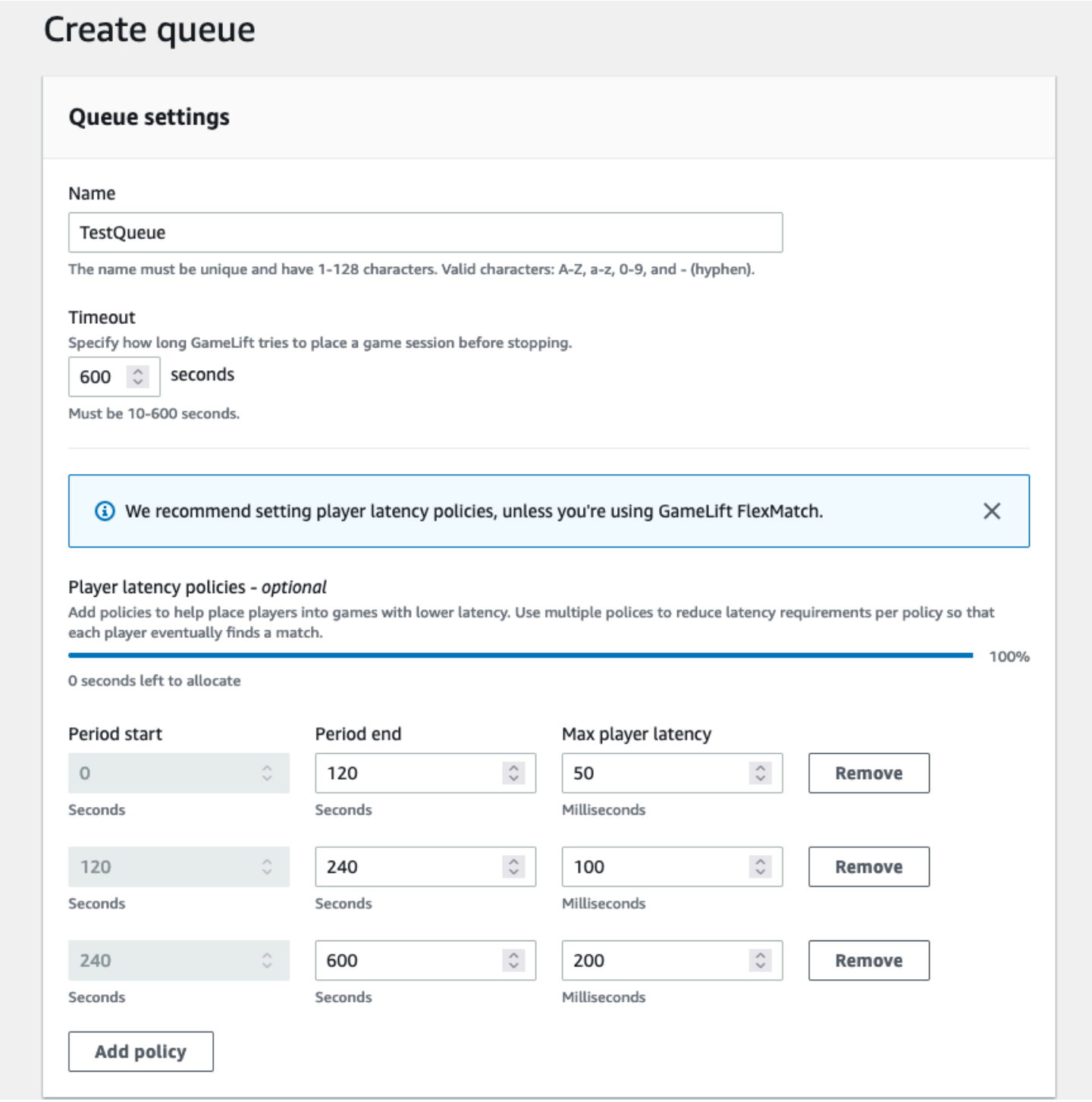
Se le tue richieste di posizionamento includono dati sulla latenza dei giocatori, Amazon GameLift Servers trova le sessioni di gioco nelle località con la latenza media più bassa per tutti i giocatori. Il posizionamento delle sessioni di gioco in base alla latenza media dei giocatori Amazon GameLift Servers impedisce di collocare la maggior parte dei giocatori in partite con latenza elevata. Tuttavia, posiziona Amazon GameLift Servers ancora i giocatori con una latenza estrema. Per soddisfare questi giocatori, crea delle politiche sulla latenza dei giocatori.

Una politica di latenza dei giocatori Amazon GameLift Servers impedisce di collocare una sessione di gioco richiesta in qualsiasi punto in cui i giocatori inclusi nella richiesta potrebbero riscontrare una latenza superiore al valore massimo. Le politiche di latenza dei giocatori possono anche Amazon GameLift Servers impedire di abbinare le richieste di sessione di gioco ai giocatori con latenza più elevata.

> **Tip**
>
> Per gestire regole specifiche sulla latenza, ad esempio richiedere una latenza simile per tutti i giocatori di un gruppo, puoi utilizzare la funzione Amazon GameLift ServersFlexMatchper creare regole di matchmaking basate sulla latenza.

Ad esempio, considera questa coda con un timeout di 5 minuti e le seguenti politiche di latenza dei giocatori:

1. Dedica 120 secondi alla ricerca di una posizione in cui tutte le latenze dei giocatori siano inferiori a 50 millisecondi.
2. Dedica 120 secondi alla ricerca di una posizione in cui tutte le latenze dei giocatori siano inferiori a 100 millisecondi.
3. Trascorri il tempo di coda rimanente fino al timeout cercando una posizione in cui tutte le latenze dei giocatori siano inferiori a 200 millisecondi.



Valuta le metriche della coda

I parametri consentono di valutare le prestazioni delle code. È possibile visualizzare le metriche relative alle code in Amazon GameLift Servers console o in Amazon CloudWatch. Per un elenco e le descrizioni delle metriche delle code, consulta. Parametri Amazon GameLift Servers per code

Le metriche della coda possono fornire informazioni su quanto segue:

- Prestazioni complessive della coda: le metriche della coda indicano il successo con cui una coda risponde alle richieste di posizionamento. Queste metriche possono anche aiutarti a identificare quando e perché i posizionamenti falliscono. Per le code con flotte ridimensionate manualmente, le `QueueDepth` metriche `AverageWaitTime` e possono indicare quando è necessario regolare la capacità di una coda.
- FleetIQ prestazioni dell'algoritmo: per le richieste di posizionamento utilizzando il FleetIQ algoritmo, le metriche mostrano la frequenza con cui l'algoritmo trova il posizionamento ideale nella sessione di gioco. Il posizionamento può dare la priorità all'utilizzo di risorse con la latenza di giocatore più bassa o di risorse con il costo più basso. Esistono anche metriche di errore che identificano i motivi più comuni Amazon GameLift Servers non riesco a trovare un posizionamento ideale. Per ulteriori informazioni sui parametri, consulta Monitora Amazon GameLift Servers con Amazon CloudWatch.
- Posizionamenti specifici per località: per le code con più sedi, le metriche mostrano i posizionamenti riusciti per località. Per le code che utilizzano il FleetIQ algoritmo, questi dati forniscono informazioni utili su dove avviene l'attività dei giocatori.

Quando si valutano le metriche per FleetIQ prestazioni dell'algoritmo, considera i seguenti suggerimenti:

- Per tenere traccia del tasso di ricerca del posizionamento ideale da parte della coda, utilizza la `PlacementsSucceeded` metrica in combinazione con FleetIQ metriche per la latenza più bassa e il prezzo più basso.
- Per aumentare il tasso di ricerca del posizionamento ideale da parte di una coda, esamina le seguenti metriche di errore:
  - Se il valore `FirstChoiceOutOfCapacity` è elevato, modifica la scalabilità della capacità per le flotte della coda.
  - Se la metrica `FirstChoiceNotViable` di errore è alta, dai un'occhiata alle flotte delle tue istanze Spot. Le flotte di istanze Spot sono considerate inutilizzabili quando il tasso di interruzione per un particolare tipo di istanza è troppo elevato. Per risolvere questo problema, modifica la coda per utilizzare flotte di istanze Spot con diversi tipi di istanze. Ti consigliamo di includere flotte di istanze Spot con diversi tipi di istanze in ogni sede.

Progettare una coda per le istanze Spot

Puoi sfruttare i notevoli risparmi sui costi di hosting utilizzando le flotte Spot. Per ulteriori dettagli, consulta Istanze On-Demand e istanze Spot. Per aggiungere flotte Spot alla tua soluzione di hosting, devi configurare una coda per le sessioni di gioco con una combinazione di flotte Spot e flotte On-Demand. Amazon GameLift Servers utilizza una coda durante il processo di posizionamento della sessione di gioco per cercare tra più flotte e trovare i migliori host disponibili per nuove sessioni di gioco. Questo argomento fornisce indicazioni su come iniziare a utilizzare le flotte Spot.

Stai usando FlexMatch per matchmaking? Puoi utilizzare i seguenti passaggi per aggiungere flotte Spot alle code delle sessioni di gioco esistenti per i posizionamenti nel matchmaking.

1. Determina le destinazioni per la coda delle sessioni di gioco.

   Gestire il posizionamento delle sessioni di gioco con una coda è la migliore pratica ed è necessaria quando si utilizzano le istanze Spot. Poiché le istanze Spot potrebbero non essere sempre disponibili quando ne hai bisogno, devi progettare una coda resiliente che includa sia flotte Spot che flotte On-Demand per offrire capacità di backup. Puoi mantenere le tue flotte On-Demand ridimensionate fino a quando non saranno necessarie. Per progettare la coda, considera quanto segue:

   - Sedi: se possibile, le tue flotte Spot e le flotte On-Demand dovrebbero trovarsi nella stessa regione dei giocatori. Posiziona sia le risorse Spot che le risorse On-Demand in ogni sede che desideri supportare. Le flotte con più sedi supportano sia le istanze Spot che quelle On-Demand.
   - Tipi di istanze: considera i requisiti hardware del tuo server di gioco e la disponibilità delle istanze nelle posizioni che scegli.

   Per provare una coda che ottimizzi la disponibilità e la resilienza di Spot, consulta. Tutorial: creare una Amazon GameLift Servers coda con le istanze Spot Per le best practice di progettazione di Spot, consulta. Procedure consigliate per le code delle sessioni di Amazon GameLift Servers gioco

2. Creare i parchi istanze per la coda con ottimizzazione Spot.

   In base al design della coda, crea flotte per distribuire i server di gioco nelle posizioni e nei tipi di istanza desiderati. Consultare Crea una EC2 flotta Amazon GameLift Servers gestita per assistenza con la creazione e la configurazione dei nuovi parchi istanze.

3. Crea la coda della tua sessione di gioco.

   Aggiungi le destinazioni della flotta, configura il processo di posizionamento delle sessioni di gioco e definisci le priorità di posizionamento. Consultare Crea una coda per le sessioni di gioco per assistenza con la creazione e la configurazione della nuova coda.

4. Aggiorna il servizio client di gioco per utilizzare la coda.

   Quando il client di gioco utilizza una coda per richiedere risorse, la coda evita le risorse con un'alta probabilità di interruzione e seleziona la posizione che corrisponde alle priorità definite. Per assistenza con l'implementazione dei posizionamenti della sessione di gioco nel client di gioco, consultare Crea sessioni di gioco.

5. Aggiorna il server di gioco per gestire un'interruzione Spot.

   AWS può interrompere le istanze Spot con una notifica di 2 minuti quando è necessario ripristinare la capacità. Configura il server di gioco in modo da gestire le interruzioni e ridurre al minimo l'impatto sui giocatori.

   Prima di AWS recuperare un'istanza Spot, invia una notifica di cessazione. Amazon GameLift Servers trasmette la notifica a tutti i processi del server interessati richiamando il Amazon GameLift Servers Funzione di callback del server SDK. `onProcessTerminate()` Implementa questa callback per terminare la sessione di gioco o spostare la sessione di gioco e i giocatori su una nuova istanza. Consultare Rispondi a una notifica di chiusura di un processo del server per assistenza con l'implementazione di `onProcessTerminate()`.

   > **Note**
   >
   > AWS compie ogni sforzo per fornire la notifica prima di recuperare un'istanza, ma è possibile che AWS recuperi l'istanza Spot prima che arrivi l'avviso. Prepara il tuo server di gioco per gestire interruzioni impreviste.

6. Controlla le prestazioni delle tue flotte e delle tue code Spot.

   Vista Amazon GameLift Servers metriche in Amazon GameLift Servers console o con Amazon CloudWatch per verificare le prestazioni. Per ulteriori informazioni sull' Amazon GameLift Servers metriche, vediMonitora Amazon GameLift Servers con Amazon CloudWatch. I parametri chiave includono:

   - Frequenza di interruzione: utilizza le `GameSessionInterruptions` metriche `InstanceInterruptions` and per tenere traccia del numero e della frequenza delle

interruzioni relative a SPOT per istanze e sessioni di gioco. Le sessioni di gioco che vengono recuperate da AWS hanno uno stato e un motivo dello stato pari a. `TERMINATED` `INTERRUPTED`

- Efficacia della coda: monitora le percentuali di successo del posizionamento, il tempo medio di attesa e la profondità della coda per verificare che le flotte Spot non influiscano sulle prestazioni in coda.
- Utilizzo della flotta: monitora i dati sulle istanze, sulle sessioni di gioco e sulle sessioni dei giocatori. L'utilizzo delle flotte On-Demand può essere un indicatore del fatto che le code impediscono il collocamento nelle flotte Spot per evitare interruzioni.

Le migliori pratiche per le code con le flotte Spot

Se la tua coda include flotte Spot, configura una coda resiliente. Ciò sfrutta i risparmi sui costi delle flotte Spot riducendo al minimo l'effetto delle interruzioni delle sessioni di gioco. Per informazioni sulla corretta creazione di flotte e code per le sessioni di gioco da utilizzare con le flotte Spot, consulta. Tutorial: creare una Amazon GameLift Servers coda con le istanze Spot Per ulteriori informazioni sulle istanze Spot, consulta. Progettare una coda per le istanze Spot

Oltre alle best practice generali della sezione precedente, prendi in considerazione queste best practice specifiche per Spot:

- Crea almeno una flotta On-Demand in ogni sede. Le flotte On-Demand forniscono server di gioco di backup per i tuoi giocatori. Puoi mantenere le tue flotte di backup ridimensionate fino a renderle necessarie e utilizzare la scalabilità automatica per aumentare la capacità on demand quando le flotte Spot non sono disponibili.
- Seleziona diversi tipi di istanze tra più flotte Spot in una posizione. Se un tipo di istanza Spot diventa temporaneamente non disponibile, l'interruzione interessa solo un parco istanze Spot nella località. La migliore pratica è scegliere tipi di istanze ampiamente disponibili e utilizzare tipi di istanze della stessa famiglia (ad esempio, m5.large, m5.xlarge, m5.2xlarge). Utilizzo dell'Amazon GameLift Servers console per visualizzare i dati storici sui prezzi per i tipi di istanza.

Imposta la notifica degli eventi per il posizionamento della sessione di gioco

Puoi utilizzare le notifiche degli eventi per monitorare lo stato delle singole richieste di collocamento. Ti consigliamo di impostare le notifiche relative agli eventi per tutti i giochi con un elevato volume di attività di posizionamento.

Sono disponibili due opzioni per impostare le notifiche di eventi.

- Avere Amazon GameLift Servers pubblica notifiche di eventi su un argomento di Amazon Simple Notification Service (Amazon SNS) utilizzando una coda.
- Usa EventBridge gli eventi Amazon pubblicati automaticamente e la sua suite di strumenti per la gestione degli eventi.

Per un elenco degli eventi di posizionamento delle sessioni di gioco emessi da Amazon GameLift Servers, consulta Eventi di collocamento delle sessioni di gioco.

Imposta un argomento SNS

In Amazon GameLift Servers per pubblicare tutti gli eventi generati da una coda di sessione di gioco su un argomento, imposta il campo di destinazione della notifica su un argomento.

Per impostare un argomento SNS per Amazon GameLift Servers event notification

1. Accedi AWS Management Console e apri la console Amazon SNS nella versione v3/home. https://console.aws.amazon.com/sns/
2. Dalla pagina Argomenti SNS, scegli Crea argomento e segui le istruzioni per creare il tuo argomento.
3. In Politica di accesso, procedi come segue:

    a. Scegli il metodo Avanzato.

    b. Aggiungi la seguente sezione in grassetto dell'oggetto JSON alla policy esistente.

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
```

```
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
        "SNS:RemovePermission",
        "SNS:DeleteTopic",
        "SNS:Subscribe",
        "SNS:ListSubscriptionsByTopic",
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:your_region:your_account:your_topic_name",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "your_account"
        }
      }
    },
    {
      "Sid": "__console_pub_0",
      "Effect": "Allow",
      "Principal": {
        "Service": "gamelift.amazonaws.com"
      },
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:your_region:your_account:your_topic_name",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn":
 "arn:aws:gamelift:your_region:your_account:gamesessionqueue/your_queue_name"
        }
      }
    }
  ]
}
```

c. (Facoltativo) Aggiungete un ulteriore controllo degli accessi all'argomento aggiungendo condizioni alla politica delle risorse.

4. Scegli Create topic (Crea argomento).

5. Dopo aver creato l'argomento SNS, aggiungilo alle code durante la creazione della coda oppure modifica una coda esistente per aggiungerlo.

Configura un argomento SNS con crittografia lato server

La crittografia lato server (SSE) consente di archiviare dati sensibili in argomenti crittografati. SSE protegge il contenuto dei messaggi negli argomenti Amazon SNS utilizzando chiavi gestite in AWS Key Management Service (AWS KMS). Per ulteriori informazioni sulla crittografia lato server con Amazon SNS, consulta Encryption at rest nella Amazon Simple Notification Service Developer Guide.

Per impostare un argomento SNS con crittografia lato server, consulta i seguenti argomenti:

- Creazione di una chiave nella Guida per gli sviluppatori AWS Key Management Service
- Abilitare SSE per un argomento nella Guida per gli sviluppatori di Amazon Simple Notification Service

Quando crei la tua chiave KMS, utilizza la seguente politica per le chiavi KMS:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "gamelift.amazonaws.com"
     },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "aws:SourceArn":
 "arn:aws:gamelift:your_region:your_account:gamesessionqueue/your_queue_name"
        },
        "StringEquals": {
            "kms:EncryptionContext:aws:sns:topicArn":
 "arn:aws:sns:your_region:your_account:your_sns_topic_name"
        }
    }
}
```

Configura EventBridge

Amazon GameLift Servers pubblica automaticamente tutti gli eventi di posizionamento delle sessioni di gioco su EventBridge. Con EventBridge puoi impostare regole per indirizzare gli eventi alle destinazioni per l'elaborazione. Ad esempio, puoi impostare una regola per indirizzare l'evento `PlacementFulfilled` a una AWS Lambda funzione che gestisce le attività che precedono la connessione a una sessione di gioco. Per ulteriori informazioni su EventBridge, consulta What is Amazon EventBridge? nella Amazon EventBridge User Guide.

Di seguito sono riportati alcuni esempi di EventBridge regole da utilizzare con Amazon GameLift Servers code:

Corrisponde agli eventi di tutti Amazon GameLift Servers code

```
{
    "source": [
        "aws.gamelift"
    ],
    "detail-type": [
        "GameLift Queue Placement Event"
    ]
}
```

Corrisponde agli eventi di una coda specifica

```
{
    "source": [
        "aws.gamelift"
    ],
    "detail-type": [
        "GameLift Queue Placement Event"
    ],
    "resources": [
        "arn:aws:gamelift:your_region:your_account:gamesessionqueue/your_queue_name"
    ]
}
```

Tutorial: creare una Amazon GameLift Servers coda con le istanze Spot

Introduzione

Questo tutorial descrive come impostare il posizionamento delle sessioni di gioco per i giochi distribuiti su flotte Spot a basso costo. Le flotte Spot richiedono passaggi aggiuntivi per mantenere la disponibilità continua dei server di gioco per i tuoi giocatori.

Destinatari principali

Questo tutorial è per gli sviluppatori di giochi che desiderano utilizzare le flotte Spot per ospitare server di gioco personalizzati o. Amazon GameLift Servers Realtime

Cosa imparerai

- Definisci il gruppo di giocatori a cui serve la coda della sessione di gioco.
- Crea un'infrastruttura di flotta per supportare l'ambito della coda delle sessioni di gioco.
- Assegna un alias a ciascuna flotta per estrarre l'ID della flotta.
- Crea una coda, aggiungi flotte e dai priorità alle sessioni di gioco. Amazon GameLift Servers
- Aggiungi politiche di latenza dei giocatori per ridurre al minimo i problemi di latenza.

Prerequisiti

Prima di creare flotte e code per il posizionamento delle sessioni di gioco, completa le seguenti attività:

- Verificare Funzionamento di Amazon GameLift Servers.
- Integra il tuo server di gioco con. Amazon GameLift Servers
- Carica la build o Realtime lo script del tuo server di gioco suAmazon GameLift Servers.
- Pianifica la configurazione della tua flotta.

Fase 1: Definisci l'ambito della coda

In questo tutorial, progettiamo una coda per un gioco con una variante di build del server di gioco. Al momento del lancio, pubblicheremo il gioco in due località: Asia Pacifico (Seoul) e Asia Pacifico (Singapore). Poiché queste località sono vicine l'una all'altra, la latenza non è un problema per i nostri giocatori.

Per questo esempio, c'è un segmento di giocatori, il che significa che creiamo una coda. In futuro, quando pubblicheremo il gioco in Nord America, potremo creare una seconda coda riservata ai giocatori nordamericani.

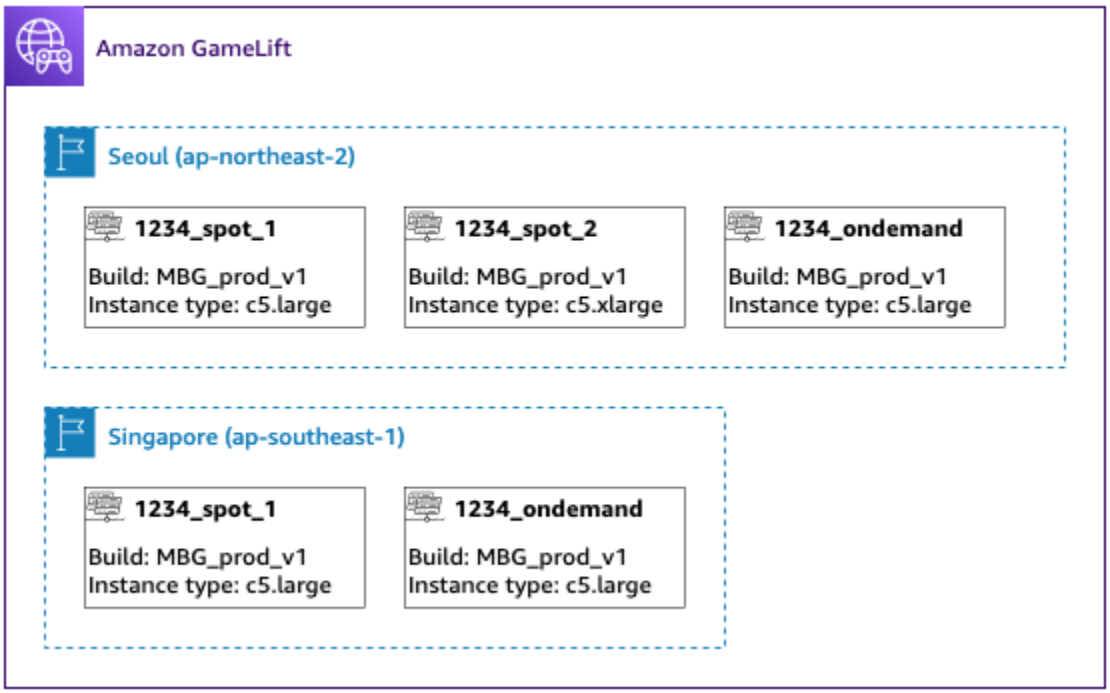
Per ulteriori informazioni, consulta Definisci l'ambito di una coda.

Fase 2: Creare l'infrastruttura della flotta Spot

Crea flotte in diverse località e con build di server di gioco o script adatti all'ambito definito. Fase 1: Definisci l'ambito della coda

In questo tutorial, creiamo un'infrastruttura a due sedi con almeno una flotta Spot e una flotta On-Demand in ciascuna sede. Ogni flotta utilizza la stessa configurazione del server di gioco. Inoltre, prevediamo che il traffico di giocatori sarà più intenso nella sede di Seoul, quindi aggiungeremo altre flotte Spot.

Il diagramma seguente mostra l'esempio di infrastruttura della flotta Spot, con 3 flotte nella località ap-northeast-2 (Seoul) e 2 flotte nella posizione ap-southeast-1 (Singapore). Tutte le istanze di entrambe le flotte utilizzano la build MBG_Prod_v1. La flotta in ap-northeast-2 contiene le seguenti configurazioni di flotta: fleet 1234_spot_1 con un tipo di istanza c5.large, fleet 1234_spot_2 con un tipo di istanza c5.xlarge e fleet 1234_ondemand con un tipo di istanza c5.large. La flotta in ap-southeast-1 contiene le seguenti configurazioni di flotta: fleet 1234_spot_1 con un tipo di istanza c5.large e fleet 1234_ondemand con un tipo di istanza c5.large.

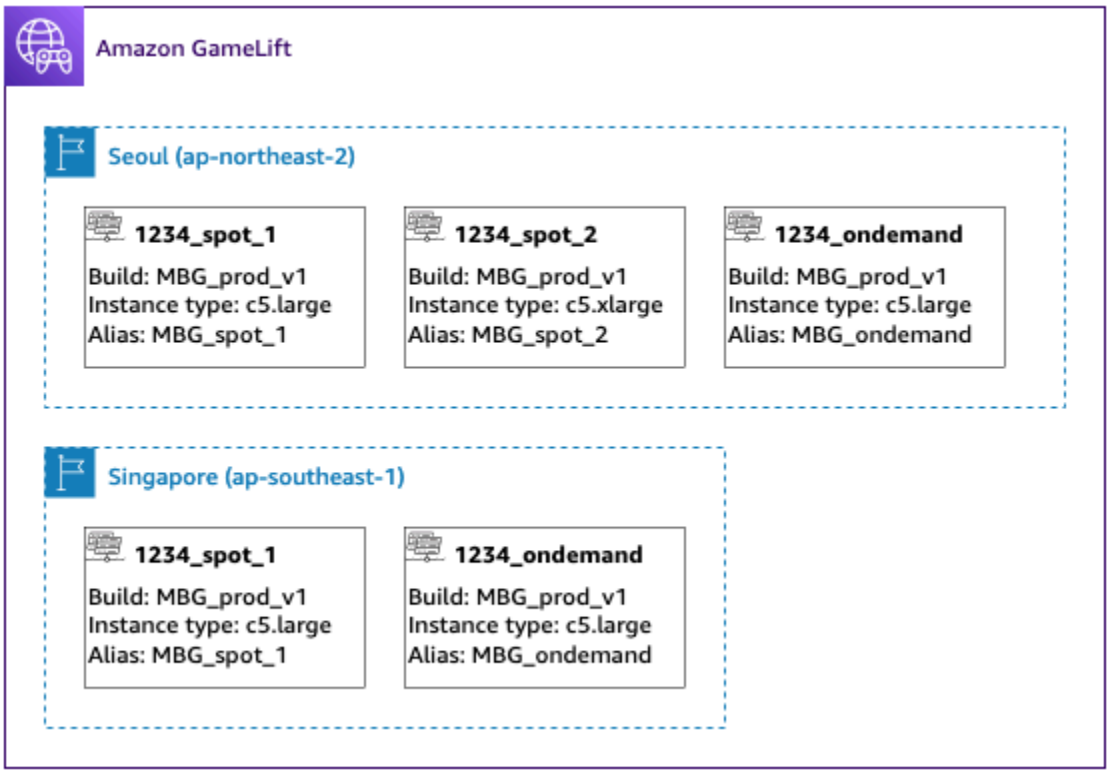


Fase 3: Assegna alias per ogni flotta

Crea un nuovo alias per ogni flotta della tua infrastruttura. Gli alias astraggono le identità della flotta, rendendo efficiente la sostituzione periodica della flotta. Per ulteriori informazioni sulla creazione di alias, consulta. Crea un Amazon GameLift Servers alias

La nostra infrastruttura di flotta ha cinque flotte, quindi creiamo cinque alias utilizzando la strategia di routing. Sono necessari tre alias nella località Asia Pacifico (Seoul) e due alias nella località Asia Pacifico (Singapore).

Il diagramma seguente mostra l'infrastruttura della flotta Spot descritta nella seconda fase con alias aggiunti a ciascuna flotta. Fleet 1234_spot_1 ha l'alias MBG_Spot_1, Fleet 1234_spot_2 ha l'alias MBG_Spot_2 e fleet 1234_ondemand ha l'alias MBG_OnDemand.



Per ulteriori informazioni, consulta Crea una coda con più sedi.

Fase 4: Creare una coda con le destinazioni

Crea la coda delle sessioni di gioco e aggiungi le destinazioni della tua flotta. Per ulteriori informazioni sulla creazione di una coda, consulta. Crea una coda per le sessioni di gioco

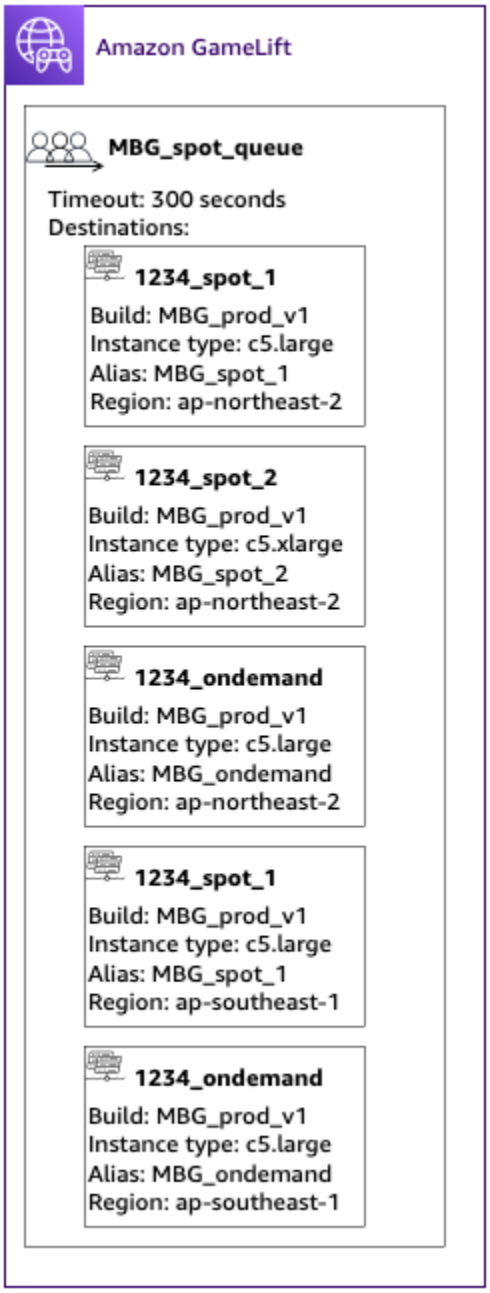
Durante la creazione della coda:

- Imposta il timeout predefinito su 10 minuti. Successivamente, potrai verificare in che modo il timeout della coda influisce sui tempi di attesa dei giocatori per accedere alle partite.
- Per ora salta la sezione sulle politiche di latenza dei giocatori. Ne parleremo nella fase successiva.
- Dai priorità alle flotte in coda. Quando lavori con le flotte Spot, consigliamo uno dei seguenti approcci:
  - Se la tua infrastruttura utilizza una sede principale con flotte in una seconda sede per il backup, dai la priorità alle flotte prima per posizione e poi per tipo di flotta.
  - Se la tua infrastruttura utilizza più sedi allo stesso modo, dai la priorità alle flotte per tipo di flotta, posizionando le flotte Spot in cima alla coda.

Per questo tutorial, creiamo una nuova coda con il nome e aggiungiamo gli alias di **`MBG_spot_queue`** tutte e cinque le nostre flotte. Quindi diamo priorità ai posizionamenti prima in base alla località e in secondo luogo in base al tipo di flotta.

In base a questa configurazione, questa coda cerca sempre di inserire nuove sessioni di gioco in una flotta Spot a Seoul. Quando queste flotte sono piene, la coda utilizza la capacità disponibile della flotta Seoul On-Demand come riserva. Se tutte e tre le flotte di Seoul non sono disponibili, Amazon GameLift Servers organizza sessioni di gioco sulle flotte di Singapore.

Il diagramma seguente mostra una coda con un timeout di 300 secondi e destinazioni prioritarie. Le destinazioni sono nell'ordine seguente: 1234_spot_1 in ap-northeast-2, 1234_spot_2 in ap-northeast-2, 1234_ondemand in ap-northeast-2, 1234_spot_1 in ap-southeast-1 e 1234_ondemand in -ap-southeast-1.



Passaggio 5: aggiungere limiti di latenza alla coda

Il nostro gioco include informazioni sulla latenza nelle richieste di posizionamento delle sessioni di gioco. Abbiamo anche una funzione player party che crea una sessione di gioco per un gruppo di giocatori. Possiamo far aspettare i giocatori un po' più a lungo prima di iniziare a giocare con l'esperienza di gioco ideale. I nostri test di gioco mostrano le seguenti osservazioni:

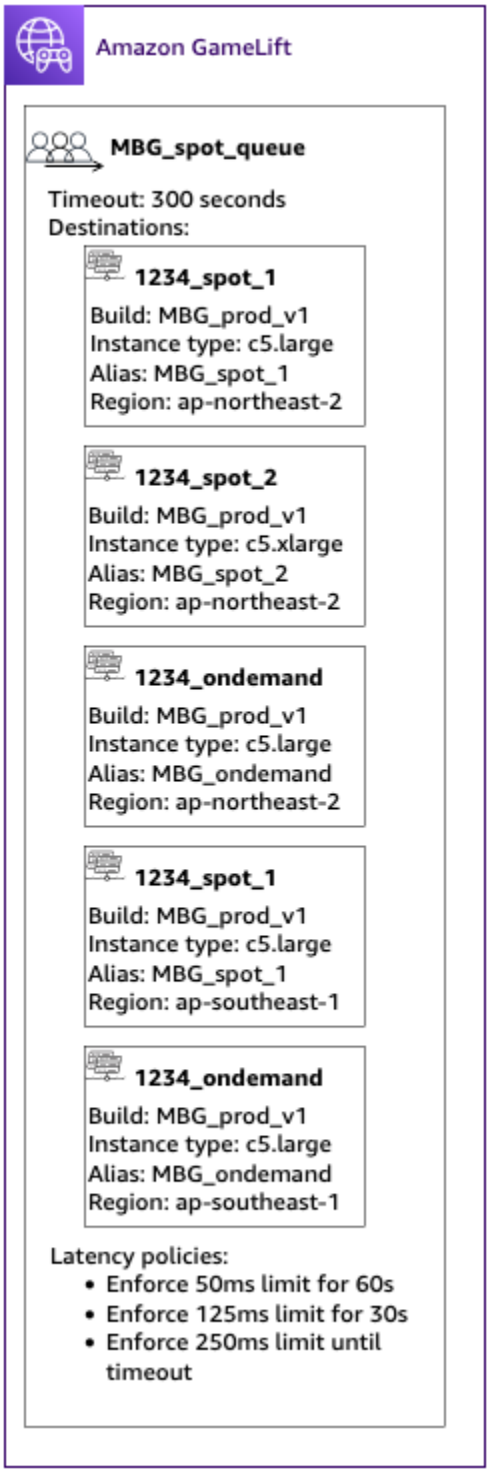
- Una latenza inferiore a 50 millisecondi è l'ideale.

- Il gioco è ingiocabile con latenze superiori a 250 millisecondi.
- I giocatori diventano impazienti dopo circa un minuto.

Per la nostra coda, con un timeout di 300 secondi, aggiungiamo delle norme che limitano la latenza consentita. Le dichiarazioni politiche consentono gradualmente valori di latenza più elevati, fino a una latenza di 250 millisecondi.

Con questa politica, la nostra coda cerca i posizionamenti con latenza ideale (inferiore a 50 millisecondi) per il primo minuto, quindi allenta il limite. La coda non crea posizionamenti in cui la latenza del giocatore è pari o superiore a 250 millisecondi.

Il diagramma seguente mostra la coda della quarta fase con l'aggiunta delle politiche di latenza dei giocatori. Le politiche di latenza del giocatore stabiliscono, applicano il limite di 50 ms per 60 secondi, impongono il limite di 125 ms per 30 secondi e impongono il limite di 250 ms fino al timeout.



Riepilogo

Complimenti! Ecco le cose che hai realizzato:

- Hai una coda per le sessioni di gioco limitata a un segmento della tua popolazione di giocatori.

- La tua coda utilizza le flotte Spot in modo efficace ed è resiliente quando si verificano interruzioni Spot.
- La tua coda dà la priorità alle flotte per garantire ai giocatori l'esperienza migliore.
- La coda ha limiti di latenza per proteggere i giocatori da esperienze di gioco negative.

Ora puoi usare la coda per organizzare sessioni di gioco per i giocatori serviti. Quando effettui richieste di posizionamento delle sessioni di gioco per questi giocatori, fai riferimento al nome della coda della sessione di gioco nella richiesta. Per ulteriori informazioni su come effettuare richieste di posizionamento per sessioni di giocoCrea sessioni di gioco, consulta oIntegrazione di un client di gioco per Amazon GameLift ServersRealtime.

Passaggi successivi:

- Progetta la tua coda.
- Crea una coda.
- Usa una coda con il tuo client di gioco.

Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers

La capacità di hosting, misurata in istanze, rappresenta il numero di sessioni di gioco che Amazon GameLift Servers possono ospitare contemporaneamente e il numero di giocatori simultanei che tali sessioni di gioco possono ospitare. Una delle attività più impegnative dell'hosting di giochi è scalare la capacità per soddisfare la domanda dei giocatori senza sprecare denaro in risorse di cui non hai bisogno. Per ulteriori informazioni, consulta Dimensionamento della capacità del parco istanze.

La capacità viene regolata a livello di ubicazione della flotta. Tutte le flotte hanno almeno una sede: la AWS regione di origine della flotta. Quando si visualizza o si aumenta la capacità, le informazioni vengono elencate per località, inclusa la regione di origine della flotta e qualsiasi località remota aggiuntiva.

Puoi impostare manualmente il numero di istanze da gestire oppure puoi impostare la scalabilità automatica per regolare dinamicamente la capacità in base alle variazioni della domanda dei giocatori. Ti consigliamo di iniziare attivando l'opzione di ridimensionamento automatico basato sulla destinazione. L'obiettivo dell'auto scaling basato sull'obiettivo è mantenere risorse di hosting sufficienti per soddisfare i giocatori attuali, oltre a un piccolo extra per gestire picchi imprevisti della domanda da parte dei giocatori. Per la maggior parte dei giochi, il ridimensionamento automatico basato sul target offre una soluzione di ridimensionamento altamente efficace.

È possibile eseguire la maggior parte delle attività di ridimensionamento del parco veicoli utilizzando la console. Amazon GameLift Servers Puoi anche utilizzare un AWS SDK o il AWS Command Line Interface (AWS CLI) con l'API del servizio per. Amazon GameLift Servers

Argomenti

- Per gestire la capacità della flotta nella console
- Imposta i limiti Amazon GameLift Servers di capacità
- Imposta manualmente la capacità per una Amazon GameLift Servers flotta
- Scalabilità automatica della capacità della flotta con Amazon GameLift Servers
- Flotte di Amazon GameLift Servers container in scala

Per gestire la capacità della flotta nella console

1. Apri la Amazon GameLift Servers console.
2. Nel riquadro di navigazione, scegli Hosting, Fleets.
3. Nella pagina Flotte, scegli il nome di una flotta attiva per aprire la pagina dei dettagli della flotta.
4. Scegli la scheda Scaling. In questa scheda puoi:
   - Visualizza le metriche di scalabilità storiche per l'intera flotta.
   - Visualizza e aggiorna le impostazioni di capacità per ogni sede del parco veicoli, inclusi i limiti di scalabilità e le impostazioni di capacità correnti.
   - Aggiorna la scalabilità automatica basata sugli obiettivi, visualizza le politiche di scalabilità automatica basate su regole applicate all'intera flotta e sospendi l'attività di auto scaling per ogni sede.

Imposta i limiti Amazon GameLift Servers di capacità

Quando scalate la capacità di hosting per una sede Amazon GameLift Servers della flotta, manualmente o tramite scalabilità automatica, considerate i limiti di scalabilità della sede. Tutte le sedi del parco veicoli hanno un limite minimo e massimo che definisce l'intervallo consentito per la capacità della sede. Per impostazione predefinita, i limiti per le sedi del parco veicoli hanno un minimo di 0 istanze e un massimo di 1 istanza. Prima di poter scalare una sede del parco veicoli, modifica i limiti.

Se utilizzi la scalabilità automatica, il limite massimo consente di ampliare la posizione di una flotta Amazon GameLift Servers per soddisfare la domanda dei giocatori, ma evita costi di hosting eccessivi, come durante un attacco DDOS. Imposta un CloudWatch allarme Amazon per avvisarti quando la capacità si avvicina al limite massimo, in modo da poter valutare la situazione e regolarla manualmente secondo necessità. (Puoi anche creare un allarme di fatturazione per monitorare AWS i costi.) Il limite minimo è utile per mantenere la disponibilità dell'hosting, anche quando la domanda dei giocatori è bassa.

Puoi impostare i limiti di capacità per le sedi di una flotta nella Amazon GameLift Serversconsole o utilizzando AWS Command Line Interface (AWS CLI).

Per impostare i limiti di capacità

Console

1. Apri la Amazon GameLift Servers console.
2. Nel pannello di navigazione, scegli Hosting, Fleets.
3. Nella pagina Flotte, scegli il nome di una flotta attiva per aprire la pagina dei dettagli della flotta.
4. Nella scheda Scalabilità, in Scalabilità della capacità, seleziona un'ubicazione della flotta, quindi scegli Modifica.
5. Nella finestra di dialogo Modifica capacità di scalabilità, imposta il numero di istanze per Dimensione minima, Istanze desiderate e Dimensione massima.
6. Scegli Conferma.

AWS CLI

1. Controllate le impostazioni di capacità correnti. In una finestra della riga di comando, usa il describe-fleet-location-capacitycomando con l'ID della flotta e la posizione per cui desideri modificare la capacità. Questo comando restituisce un FleetCapacityoggetto che include le impostazioni di capacità correnti della posizione. Determina se i nuovi limiti di istanza possono soddisfare l'attuale impostazione delle istanze desiderate.

```
aws gamelift describe-fleet-location-capacity \
    --fleet-id <fleet identifier> \
    --location <location name>
```

2. Update limit settings (Aggiornare le impostazioni dei limiti). In una finestra della riga di comando, utilizzate il update-fleet-capacitycomando con i seguenti parametri. È possibile regolare i limiti dell'istanza e il conteggio desiderato delle istanze con lo stesso comando.

```
--fleet-id <fleet identifier>
--location <location name>
--max-size <maximum capacity for scaling>
--min-size <minimum capacity for scaling>
--desired-instances <fleet capacity goal>
```

Esempio:

```
aws gamelift update-fleet-capacity \
    --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \
    --location us-west-2 \
    --max-size 10 \
    --min-size 1 \
    --desired-instances 10
```

Se la richiesta ha esito positivo, Amazon GameLift Servers restituisce l'ID della flotta. Se il nuovo `min-size` valore `max-size` o il valore è in conflitto con l'`desired-instances`impostazione corrente, Amazon GameLift Servers restituisce un errore.

Imposta manualmente la capacità per una Amazon GameLift Servers flotta

Quando crei un nuovo parco istanze, imposta Amazon GameLift Servers automaticamente le istanze desiderate su un'istanza per ciascuna ubicazione del parco istanze. Quindi, Amazon GameLift Servers distribuisce una nuova istanza in ogni posizione. Per modificare la capacità della flotta, puoi aggiungere una politica di scalabilità automatica basata sull'obiettivo oppure puoi impostare manualmente il numero di istanze che desideri per una sede. Per ulteriori informazioni, consulta Dimensionamento della capacità del parco istanze.

L'impostazione manuale della capacità di una flotta può essere utile quando non è necessaria la scalabilità automatica o quando è necessario mantenere la capacità a un livello specifico. L'impostazione manuale della capacità funziona solo se non si utilizza una politica di auto scaling basata sulla destinazione. Se disponi di una politica di scalabilità automatica basata sulla destinazione, ripristina immediatamente la capacità desiderata in base alle proprie regole di scalabilità.

È possibile impostare manualmente la capacità nella Amazon GameLift Servers console o utilizzando (). AWS Command Line Interface AWS CLI Lo stato della flotta deve essere attivo.

Sospendere il ridimensionamento automatico

Puoi sospendere tutte le attività di autoscaling per ogni sede del parco veicoli. Con la scalabilità automatica sospesa, il numero desiderato di istanze nell'ubicazione del parco macchine rimane lo stesso, a meno che non venga modificato manualmente. Quando sospendi la scalabilità automatica

per una sede, ciò influisce sulle politiche attuali del parco veicoli e su eventuali politiche che potresti definire in futuro.

Per impostare manualmente la capacità del parco istanze

Console

1. Apri la Amazon GameLift Servers console.
2. Nel pannello di navigazione, scegli Hosting, Fleets.
3. Nella pagina Flotte, scegli il nome di una flotta attiva per aprire la pagina dei dettagli della flotta.
4. Nella scheda Ridimensionamento, in Posizioni di auto-scaling sospese, seleziona ogni posizione per cui desideri sospendere il ridimensionamento automatico, quindi scegli Sospendi.
5. In Capacità di scalabilità, seleziona una posizione che desideri impostare manualmente, quindi scegli Modifica.
6. Nella finestra di dialogo Modifica capacità di scalabilità, imposta il valore preferito per Istanze desiderate, quindi scegli Conferma. Indica Amazon GameLift Servers il numero di istanze da mantenere attive, pronte per ospitare sessioni di gioco.

Amazon GameLift Serversrisponde alle modifiche distribuendo istanze aggiuntive o chiudendo quelle non necessarie. Amazon GameLift ServersAl termine di questo processo, il numero di istanze attive nella posizione cambia per corrispondere al valore aggiornato delle istanze desiderate. Questo processo può richiedere alcuni istanti.

AWS CLI

1. Controllate le impostazioni di capacità correnti. In una finestra della riga di comando, usa il describe-fleet-location-capacitycomando con l'ID della flotta e la posizione per cui desideri modificare la capacità. Questo comando restituisce un FleetCapacityoggetto che include le impostazioni di capacità correnti della posizione. Determina se i limiti delle istanze possono soddisfare la nuova impostazione delle istanze desiderata.

```
aws gamelift describe-fleet-location-capacity \
    --fleet-id <fleet identifier> \
    --location <location name>
```

2. Aggiornare la capacità desiderata. Utilizza il update-fleet-capacitycomando con l'ID della flotta, la posizione e un nuovo valore per le istanze desiderate. Se questo valore non rientra nell'intervallo limite corrente, puoi regolare i valori limite con lo stesso comando.

```
--fleet-id <fleet identifier>
--location <location name>
--desired-instances <fleet capacity as an integer>
--max-size <maximum capacity>    [Optional]
--min-size <minimum capacity>    [Optional]
```

Esempio:

```
aws gamelift update-fleet-capacity \
    --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \
    --location us-west-2 \
    --desired-instances 5 \
    --max-size 10 \
    --min-size 1
```

Se la richiesta ha esito positivo, Amazon GameLift Servers restituisce l'ID della flotta. Se la nuova impostazione delle istanze desiderate non rientra nei limiti minimo e massimo, Amazon GameLift Servers restituisce un errore.

Scalabilità automatica della capacità della flotta con Amazon GameLift Servers

Usa l'auto-scaling in Amazon GameLift Servers per scalare dinamicamente la capacità della tua flotta in risposta all'attività dei server di gioco. Man mano che i giocatori arrivano e iniziano le sessioni di gioco, la scalabilità automatica può aggiungere altre istanze; man mano che la domanda dei giocatori diminuisce, la scalabilità automatica può terminare le istanze non necessarie. La scalabilità automatica è un modo efficace per ridurre al minimo le risorse e i costi di hosting, garantendo al contempo un'esperienza di gioco fluida e veloce.

Per utilizzare la scalabilità automatica, si creano politiche di ridimensionamento che indicano Amazon GameLift Servers quando aumentare o diminuire. Esistono due tipi di politiche di scalabilità: basate sugli obiettivi e basate su regole. L'approccio basato sugli obiettivi, il monitoraggio degli obiettivi, è una soluzione completa. La consigliamo perché è l'opzione più semplice ed efficace. Le politiche di

scalabilità basate su regole richiedono la definizione di ogni aspetto del processo decisionale relativo alla scalabilità automatica, utile per risolvere problemi specifici. Questa soluzione funziona al meglio come complemento alla scalabilità automatica basata sull'obiettivo.

È possibile gestire la scalabilità automatica basata sull'obiettivo utilizzando Amazon GameLift Servers console, AWS Command Line Interface (AWS CLI) o un SDK. AWS È possibile gestire la scalabilità automatica basata su regole utilizzando solo l'SDK AWS CLI o un AWS SDK, sebbene sia possibile visualizzare le politiche di scalabilità basate su regole nella console.

Argomenti

Scalabilità automatica basata sull'obiettivo

Scalabilità automatica basata sull'obiettivo per Amazon GameLift Servers regola i livelli di capacità in base alla metrica della flotta. `PercentAvailableGameSessions` Questa metrica rappresenta il buffer disponibile della flotta per aumenti improvvisi della domanda dei giocatori.

Il motivo principale per mantenere un buffer di capacità è il tempo di attesa di un giocatore. Quando gli slot per le sessioni di gioco sono pronti e in attesa, bastano pochi secondi per coinvolgere nuovi giocatori nelle sessioni di gioco. Se non ci sono risorse disponibili, i giocatori devono attendere la fine delle sessioni di gioco esistenti o la disponibilità di nuove risorse. L'avvio di nuove istanze e processi del server può richiedere alcuni minuti.

Quando configuri la scalabilità automatica basata sull'obiettivo, specifica la dimensione del buffer che desideri venga gestita dalla flotta. Poiché `PercentAvailableGameSessions` misura la percentuale di risorse disponibili, la dimensione effettiva del buffer è una percentuale della capacità totale del parco veicoli. Amazon GameLift Servers aggiunge o rimuove istanze per mantenere la dimensione del buffer di destinazione. Con un buffer di grandi dimensioni, riduci al minimo i tempi di attesa, ma paghi anche per risorse extra che potresti non utilizzare. Se i giocatori sono più tolleranti rispetto ai tempi di attesa, è possibile ridurre i costi impostando un piccolo buffer.

Per impostare la scalabilità automatica basata sull'obiettivo

Console

1. Aprire Amazon GameLift Servers console.

2. Nel pannello di navigazione, scegli Hosting, Fleets.
3. Nella pagina Flotte, scegli il nome di una flotta attiva per aprire la pagina dei dettagli della flotta.
4. Scegli la scheda Scaling. Questa scheda mostra i parametri del dimensionamento storico del parco istanze e contiene i controlli per modificare le impostazioni di dimensionamento correnti.
5. In Capacità di scalabilità, verifica che i limiti di dimensione minima e dimensione massima siano appropriati per il parco veicoli. Con la scalabilità automatica abilitata, la capacità si regola tra questi due limiti.
6. Nella politica di auto-scaling basata su Target, scegli Modifica.
7. Nella finestra di dialogo Modifica politica di auto-scaling basata sugli obiettivi, per Percentuale di sessioni di gioco disponibili, imposta la percentuale che desideri mantenere, quindi scegli Conferma. Dopo aver confermato le impostazioni, Amazon GameLift Servers aggiunge una nuova politica basata sulla destinazione nella politica di auto-scaling basata su Target.

AWS CLI

1. Impostare i limiti di capacità. Imposta i valori limite utilizzando il comando. update-fleet-capacity Per ulteriori informazioni, consulta Imposta i limiti Amazon GameLift Servers di capacità.
2. Creare una nuova policy. Apri una finestra della riga di comando e usa il put-scaling-policycomando con le impostazioni dei parametri della tua politica. Per aggiornare una policy esistente, specificare il nome della policy e fornire una versione completa della policy aggiornata.

```
--fleet-id <unique fleet identifier>
--name "<unique policy name>"
--policy-type <target- or rule-based policy>
--metric-name <name of metric>
--target-configuration <buffer size>
```

Esempio:

```
aws gamelift put-scaling-policy \
    --fleet-id "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa" \
    --name "My_Target_Policy_1" \
    --policy-type "TargetBased" \
```

```
    --metric-name "PercentAvailableGameSessions" \
    --target-configuration "TargetValue=5"
```

Scalabilità automatica con policy basate su regole

Le politiche di scalabilità basate su regole Amazon GameLift Servers forniscono un controllo preciso quando si ridimensiona automaticamente la capacità di una flotta in risposta all'attività dei giocatori. Per ogni policy, puoi collegare la scalabilità a una delle diverse metriche della flotta, identificare un punto di attivazione e personalizzare l'evento di scalabilità verticale o discendente che risponde. Le politiche basate su regole sono utili per integrare la scalabilità basata sugli obiettivi per gestire circostanze speciali.

Una politica basata su regole stabilisce quanto segue: «Se una metrica della flotta soddisfa o supera un valore di soglia per un certo periodo di tempo, modifica la capacità della flotta di un importo specificato». Questo argomento descrive la sintassi utilizzata per un'istruzione di policy e offre assistenza nella creazione e nella gestione di policy basate su regole.

Gestire policy basate su regole

Crea, aggiorna o elimina politiche basate su regole utilizzando un AWS SDK o AWS Command Line Interface ()AWS CLI con l'API di servizio per. Amazon GameLift Servers Le policy attive possono essere visualizzate nella console Amazon GameLift Servers.

Per interrompere temporaneamente tutte le politiche di scalabilità per una flotta, usa il comando. AWS CLI stop-fleet-actions

Per creare o aggiornare una politica di scalabilità basata su regole ():AWS CLI

1. Impostare i limiti di capacità. Imposta uno o entrambi i valori limite utilizzando il comando. update-fleet-capacity Per ulteriori informazioni, consulta Imposta i limiti Amazon GameLift Servers di capacità.
2. Creare una nuova policy. Apri una finestra della riga di comando e usa il put-scaling-policycomando con le impostazioni dei parametri della tua politica. Per aggiornare una policy esistente, specificare il nome della policy e fornire una versione completa della policy aggiornata.

```
--fleet-id <unique fleet identifier>
--name "<unique policy name>"
--policy-type <target- or rule-based policy>
--metric-name <name of metric>
```

```
--comparison-operator <comparison operator>
--threshold <threshold integer value>
--evaluation-periods <number of minutes>
--scaling-adjustment-type <adjustment type>
--scaling-adjustment <adjustment amount>
```

Esempio:

```
aws gamelift put-scaling-policy \
    --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \
    --name "Scale up when AGS<50" \
    --policy-type RuleBased \
    --metric-name AvailableGameSessions \
    --comparison-operator LessThanThreshold \
    --threshold 50 \
    --evaluation-periods 10 \
    --scaling-adjustment-type ChangeInCapacity \
    --scaling-adjustment 1
```

Per eliminare una politica di scalabilità basata su regole utilizzando: AWS CLI

- Aprire una finestra della riga di comando e utilizzare il delete-scaling-policycomando con l'ID della flotta e il nome della politica.

  Esempio:

```
aws gamelift delete-scaling-policy \
    --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \
    --name "Scale up when AGS<50"
```

Sintassi per le regole di ridimensionamento automatico

Per creare una dichiarazione di politica di scalabilità basata su regole, specifica sei variabili:

Se *<metric name>* rimane *<comparison operator>* *<threshold value>* per*<evaluation period>*, modifica la capacità della flotta utilizzando. *<adjustment type>* to/by *<adjustment value>*

Ad esempio, questa dichiarazione politica avvia un evento di ampliamento ogni volta che la capacità aggiuntiva di una flotta è inferiore a quella necessaria per gestire 50 nuove sessioni di gioco:

Se `AvailableGameSessions` rimane a `less than 50` per `10 minutes`, utilizzare `ChangeInCapacity` per modificare la capacità del parco istanze di `1 instances`.

Nome parametro

Per avviare un evento di scalabilità, collega una politica di scalabilità automatica a una delle seguenti metriche specifiche del parco veicoli. Per le descrizioni complete delle metriche, consulta. Parametri di Amazon GameLift Servers per parchi istanze

- Attivazione delle sessioni di gioco
- Sessioni di gioco attive
- Sessioni di gioco disponibili
- Percentuale di sessioni di gioco disponibili
- Istanze attive
- Sessioni giocatore disponibili
- Sessioni giocatore correnti
- Istanze inattive
- Percentuale di istanze inattive

Se la flotta è in coda per una sessione di gioco, puoi utilizzare le seguenti metriche:

- Profondità della coda: il numero di richieste di sessioni di gioco in sospeso per cui questa flotta è la migliore posizione di hosting disponibile.
- Tempo di attesa: tempo di attesa specifico per la flotta. L'attesa necessaria per il completamento della richiesta di sessione di gioco in attesa meno recente. Il tempo di attesa di un parco istanze corrisponde al tempo in coda della richiesta corrente meno recente.

Operatore di confronto

Indica Amazon GameLift Servers come confrontare i dati della metrica con il valore di soglia. Gli operatori di confronto validi includono maggiore di (>), minore di (<), greater than or equal (> =) e minore o uguale (< =).

Valore di soglia

Quando il valore della metrica specificato raggiunge o supera il valore di soglia, avvia un evento di scala. Questo valore è sempre un numero intero positivo.

Periodo di valutazione

La metrica deve soddisfare o superare il valore di soglia per l'intera durata del periodo di valutazione prima di iniziare un evento di scalabilità. La lunghezza del periodo di valutazione è consecutiva: se il parametro scende al di sotto del valore di soglia, il periodo di valutazione inizia nuovamente.

Tipo e valore di adeguamento

Questo set di variabili collabora per Amazon GameLift Servers specificare come regolare la capacità della flotta all'inizio di un evento di scalabilità. Scegliete tra tre possibili tipi di regolazione:

- Modifica della capacità: aumenta o diminuisce la capacità attuale di un numero specifico di istanze. Il valore di adeguamento deve essere impostato sul numero di istanze da aggiungere o rimuovere dal parco istanze. I valori positivi aggiungono istanze, mentre quelli negativi le rimuovono. Ad esempio, un valore di «-10" riduce il parco istanze di 10 istanze, indipendentemente dalla dimensione totale del parco istanze.
- Variazione percentuale della capacità: aumenta o diminuisce la capacità attuale di una percentuale specificata. Imposta il valore di regolazione sulla percentuale di cui desideri aumentare o diminuire la capacità della flotta. I valori positivi aggiungono istanze, mentre quelli negativi le rimuovono. Ad esempio, per un parco istanze con 50 istanze, una variazione percentuale di «20" aggiunge 10 istanze al parco istanze.
- Capacità esatta: aumenta o diminuisce la capacità attuale fino a un valore specifico. Il valore di adeguamento deve essere impostato sul numero esatto di istanze che si desidera mantenere nel parco istanze.

Suggerimenti per il ridimensionamento automatico basato su regole

I seguenti suggerimenti possono aiutarti a ottenere il massimo dalla scalabilità automatica con policy basate su regole.

Utilizzare più policy

Puoi avere più politiche di scalabilità automatica per una flotta contemporaneamente. Lo scenario più comune consiste in una policy mirata che gestisce la maggior parte delle esigenze di dimensionamento e utilizza le policy basate su regole per gestire i casi limite. Non ci sono limiti all'utilizzo di più politiche.

Con più policy, ogni policy si comporta in modo indipendente. Non è possibile controllare la sequenza degli eventi di scalabilità. Ad esempio, se hai più politiche che favoriscono la scalabilità, è possibile

che l'attività dei giocatori dia inizio a più eventi di scalabilità contemporaneamente. Evita politiche che si avviano l'una con l'altra. Ad esempio, è possibile creare un ciclo infinito se si creano policy scalabili verso l'alto e verso il basso che impostano la capacità oltre la soglia reciproca.

Impostare la capacità minima e massima

Ogni parco istanze ha un limite di capacità massimo e uno minimo. Questa funzionalità è importante quando si utilizza il ridimensionamento automatico. La scalabilità automatica non imposta mai la capacità su un valore al di fuori di questo intervallo. Per impostazione predefinita, i parchi istanze di nuova creazione hanno come limite minimo 0 e come limite massimo 1. Affinché la politica di scalabilità automatica influisca sulla capacità come previsto, aumentate il valore massimo.

La capacità del parco veicoli è inoltre limitata dai limiti relativi al tipo di istanza del parco macchine e dalle quote di servizio previste per ciascun parco macchine. Account AWS Non puoi impostare un minimo e un massimo al di fuori di questi limiti e quote di account.

Monitorare i parametri dopo una modifica della capacità

Dopo aver modificato la capacità in risposta a una policy di auto scaling, Amazon GameLift Servers attende 10 minuti prima di rispondere ai trigger della stessa policy. Questa Amazon GameLift Servers attesa consente di aggiungere nuove istanze, avviare i server di gioco, connettere i giocatori e iniziare a raccogliere dati dalle nuove istanze. Durante questo periodo, Amazon GameLift Servers valuta la policy rispetto alla metrica e tiene traccia del periodo di valutazione della policy, che riprende dopo che si verifica un evento di scalabilità. Ciò significa che una politica di scalabilità potrebbe avviare un altro evento di scalabilità subito dopo il termine del tempo di attesa.

Non vi è alcun tempo di attesa tra gli eventi di scalabilità che iniziano le diverse politiche di auto scaling.

Flotte di Amazon GameLift Servers container in scala

Una delle attività più impegnative dell'hosting di giochi è scalare la capacità per soddisfare la domanda dei giocatori senza sprecare denaro in risorse di cui non hai bisogno. In una flotta di container gestita, puoi scalare la capacità della flotta aggiungendo o rimuovendo istanze della flotta.

Quando crei una nuova flotta, Amazon GameLift Servers imposta la capacità desiderata della flotta su un'istanza e distribuisce un'istanza nella regione di origine della flotta. Per un parco veicoli con più sedi, Amazon GameLift Servers distribuisce un'istanza nella regione di origine e in ogni postazione

remota. Una volta raggiunto lo stato della flotta`ACTIVE`, è possibile aumentare la capacità desiderata per aumentare o ridurre la capacità desiderata di ridimensionamento.

Puoi utilizzare le funzionalità Amazon GameLift Servers di scalabilità per modificare la capacità manualmente o impostare la scalabilità automatica in base alla domanda dei giocatori:

- Imposta il ridimensionamento automatico con tracciamento del target. Consultare Scalabilità automatica basata sull'obiettivo.
- Modifica manualmente la capacità della tua flotta. Consultare Imposta manualmente la capacità per una Amazon GameLift Servers flotta.

Quando ridimensionate una flotta di container, considerate l'impatto dell'aggiunta o della rimozione di istanze sulla capacità della flotta di ospitare sessioni di gioco e giocatori.

- Sessioni di gioco per istanza
  - Ogni processo del server di gioco in esecuzione su un'istanza rappresenta la capacità di ospitare una sessione di gioco.
  - Usa questa formula per calcolare il numero di sessioni di gioco eseguite contemporaneamente su un'istanza di flotta di container:

    ```
    [Game sessions per instance] = [# of game server processes per game server
     container] * [# of game server container groups per instance]
    ```

    Se l'architettura del contenitore esegue contemporaneamente un processo del server di gioco nel contenitore del server di gioco, le sessioni di gioco per istanza equivalgono al numero di gruppi di contenitori di server di gioco per istanza.

    - Per i gruppi di contenitori di server di gioco per esempio, chiama DescribeContainerFleetper ottenere il `MaximumGameServerContainerGroupsPerInstance` valore `GameServerContainerGroupsPerInstance` or.
- Giocatori per esempio
  - Sei tu a decidere il numero di slot per i giocatori da consentire in ogni sessione di gioco. A seconda di come la tua soluzione di hosting gestisce il posizionamento delle sessioni di gioco, puoi definire i giocatori per sessione di gioco nella configurazione di matchmaking o nelle chiamate per avviare una sessione di gioco.
  - Usa questa formula per calcolare il numero di giocatori che possono giocare contemporaneamente su un'istanza di flotta di container:

```
[Players per instance] = [# of game sessions per instance] * [# of player slots per
 game session]
```

Per ottenere l'attuale capacità totale di una flotta di container, chiama DescribeFleetCapacityo DescribeFleetLocation Capacity per ottenere il numero di gruppi di container di server di gioco presenti nella flotta. I gruppi attivi sono quelli che attualmente ospitano sessioni di gioco. I gruppi inattivi sono pronti a ospitare una nuova sessione di gioco. Moltiplica questi valori per il numero di processi server per gruppo di contenitori di server di gioco.

Preparazione del gioco per il lancio con Amazon GameLift Servers hosting

Usa le seguenti liste di controllo per convalidare ogni fase di implementazione del gioco. Gli elementi contrassegnati come [Critici] sono fondamentali per il lancio della produzione.

Scarica e completa il questionario di Amazon GameLift Servers lancio, disponibile nella Amazon GameLift Servers console. Desideriamo che ogni sviluppatore di giochi abbia una giornata di lancio senza intoppi e le informazioni richieste ci aiutano a prepararti per i prossimi test di carico, il soft launch o il lancio pubblico. Amazon GameLift Servers Pianifica di inviare il questionario compilato almeno tre (3) mesi prima del primo test di carico.

Argomenti

Prepara il gioco

- [Critico] Verifica di aver completato tutti i passaggi della roadmap di sviluppo della tua soluzione di hosting e di disporre di tutti i componenti necessari, tra cui un server di gioco integrato, un servizio di backend per client di gioco, flotte di hosting e un metodo di posizionamento delle sessioni di gioco (ad esempio una coda).
- [Critico] Crea ruoli AWS Identity and Access Management (IAM) che consentano al server di gioco di accedere ad altre AWS risorse durante l'esecuzione.
- [Critico] Progetta e implementa il failover su altre risorse di hosting in base alle esigenze.
- Pianifica l'implementazione delle flotte verso le località di destinazione, considerando la coda di gioco e la struttura della flotta.
- Automatizza la distribuzione utilizzando Infrastructure as Code (IaC) con e il. AWS CloudFormation AWS Cloud Development Kit (AWS CDK)
- Raccogli log e analisi utilizzando Amazon CloudWatch e Amazon Simple Storage Service (Amazon S3).

Preparati per il test

- [Critico] Richiedi aumenti per le quote di Amazon GameLift Servers servizio e altre Servizio AWS quote in modo che il tuo ambiente live possa adattarsi alle esigenze di produzione.
- [Critico] Verificate che le porte aperte sulle flotte attive corrispondano alla gamma di porte che i server potrebbero utilizzare.
- [Critico] Chiudi la porta RDP 3389 e la porta SSH 22.
- Sviluppa un piano per la DevOps gestione del gioco. Se utilizzi Amazon CloudWatch Logs o i parametri CloudWatch personalizzati di Amazon, definisci gli allarmi per problemi gravi o critici sulla flotta di server. Simula gli errori e testa i runbook.
- Verifica che le risorse di elaborazione che stai utilizzando siano in grado di supportare il numero di processi server che desideri eseguire contemporaneamente su ogni elaborazione.
- Ottimizza la tua politica di scalabilità in modo che sia inizialmente più conservativa e fornisca più capacità inattiva di quella che ritieni necessaria. Puoi ottimizzare i costi in un secondo momento. Prendi in considerazione l'utilizzo di una politica di scalabilità basata sugli obiettivi con una capacità inattiva del 20%.
- Ad esempioFlexMatch, usa le regole di latenza per abbinare giocatori geograficamente vicini l'uno all'altro. Verifica come si comporta sotto carico con i dati di latenza sintetici del tuo client di test di carico.
- Esegui il test di caricamento dell'infrastruttura di autenticazione dei giocatori e delle sessioni di gioco per vedere se è scalabile in modo efficace per soddisfare la domanda.
- Verifica che un server rimasto in funzione per diversi giorni possa ancora accettare connessioni.
- Innalza il livello del Supporto piano a Business o Enterprise in modo che AWS possa risponderti in caso di problemi o interruzioni.

Preparati per il lancio

- [Critico] Imposta la politica di protezione della flotta sulla protezione completa di tutte le flotte attive in modo che il ridimensionamento non interrompa le sessioni di gioco attive.
- [Critico] Imposta le dimensioni massime della flotta sufficientemente elevate da soddisfare al minimo i picchi di domanda previsti. Ti consigliamo di raddoppiare la dimensione massima in caso di domanda imprevista.
- Incoraggia l'intero team di sviluppo a partecipare all'evento di lancio e a monitorare il lancio del gioco in una sala lancio.

- Monitora la latenza e l'esperienza dei giocatori.

Pianifica gli aggiornamenti post-lancio

- Ottimizza la politica di scalabilità per ridurre al minimo la capacità inattiva in base all'utilizzo del giocatore.
- Modifica FlexMatch le regole o aggiungi sedi di hosting in base ai dati sulla latenza dei giocatori e ai requisiti rivisti.
- Ottimizza la configurazione di runtime per eseguire quante più sessioni di gioco possibile su ciascuna risorsa di elaborazione. Massimizzare l'efficienza delle prestazioni in questo modo può influire direttamente sui costi del parco macchine, perché potresti essere in grado di eseguire più processi server con le stesse risorse di elaborazione.
- Usa i tuoi dati di analisi per promuovere lo sviluppo continuo, migliorare l'esperienza dei giocatori e la longevità del gioco e ottimizzare la monetizzazione.

Gestione di Amazon GameLift Servers hosting di risorse utilizzando AWS CloudFormation

Puoi usare AWS CloudFormation per gestire il tuo Amazon GameLift Servers risorse. In AWS CloudFormation, crei un modello che modella ogni risorsa e poi usi il modello per creare le tue risorse. Per aggiornare le risorse, apporti le modifiche al modello e le utilizzi AWS CloudFormation per implementare gli aggiornamenti. È possibile organizzare le risorse in gruppi logici, chiamati stack e set di stack.

Utilizzato AWS CloudFormation per mantenere il tuo Amazon GameLift Servers le risorse di hosting offrono un modo più efficiente per gestire set di AWS risorse. Puoi utilizzare il controllo della versione per tenere traccia delle modifiche dei modelli nel corso tempo e coordinare gli aggiornamenti effettuati da più membri del team. È inoltre possibile riutilizzare i modelli. Ad esempio, quando distribuisci un gioco in più regioni, puoi utilizzare lo stesso modello per creare risorse identiche in ogni regione. È inoltre possibile utilizzare questi modelli per distribuire gli stessi set di risorse in un'altra partizione.

Per ulteriori informazioni in merito AWS CloudFormation, consulta la Guida AWS CloudFormation per l'utente. Per visualizzare le informazioni sul modello per Amazon GameLift Servers risorse, vedi il Amazon GameLift Servers riferimento al tipo di risorsa.

Best practice

Per una guida dettagliata sull'utilizzo AWS CloudFormation, consulta le AWS CloudFormation migliori pratiche nella Guida AWS CloudFormation per l'utente. Inoltre, queste best practice hanno particolare rilevanza per Amazon GameLift Servers.

- Gestisci in modo coerente le tue risorse tramite AWS CloudFormation. Se modifichi le tue risorse al di fuori delle AWS CloudFormation tue risorse, le tue risorse non saranno sincronizzate con i tuoi modelli di risorse.
- Usa AWS CloudFormation pile e set di stack per gestire in modo efficiente più risorse.
  - Usa gli stack per gestire gruppi di risorse connesse. Ad esempio, uno stack che contiene una build, una flotta che fa riferimento alla build e un alias che fa riferimento alla flotta. Se aggiorni il modello per sostituire una build, AWS CloudFormation sostituisce le flotte collegate alla build. AWS CloudFormation quindi aggiorna gli alias esistenti in modo che puntino alle nuove flotte. Per ulteriori informazioni, consultate Working with stacks nella Guida per l'AWS CloudFormation utente.

  - Usa i set di AWS CloudFormation stack se stai distribuendo stack identici in più regioni o account. AWS Per ulteriori informazioni, consulta Lavorare con i set di stack nella Guida per l'utente.AWS CloudFormation
- Se utilizzi istanze Spot, includi un parco istanze on demand come backup. Ti consigliamo di configurare i tuoi modelli con due flotte in ogni regione, un parco con istanze Spot e un parco con istanze On-Demand.
- Raggruppa le risorse specifiche della località e le risorse globali in pile separate quando gestisci risorse in più sedi.
- Posiziona le tue risorse globali vicino ai servizi che le utilizzano. Risorse come le code e le configurazioni di matchmaking tendono a ricevere un volume elevato di richieste da fonti specifiche. Posizionando le risorse vicino alla fonte di tali richieste, riduci al minimo i tempi di viaggio delle richieste e puoi migliorare le prestazioni complessive.
- Posiziona la configurazione dell'abbinamento nella stessa regione della coda delle sessioni di gioco utilizzata.
- Crea un alias separato per ogni parco istanze nello stack.

Utilizzo delle AWS CloudFormation pile

Consigliamo di utilizzare le seguenti strutture per la configurazione degli AWS CloudFormation stack per Amazon GameLift Servers risorse. La struttura ottimale dello stack varia a seconda che il gioco venga distribuito in una o più postazioni.

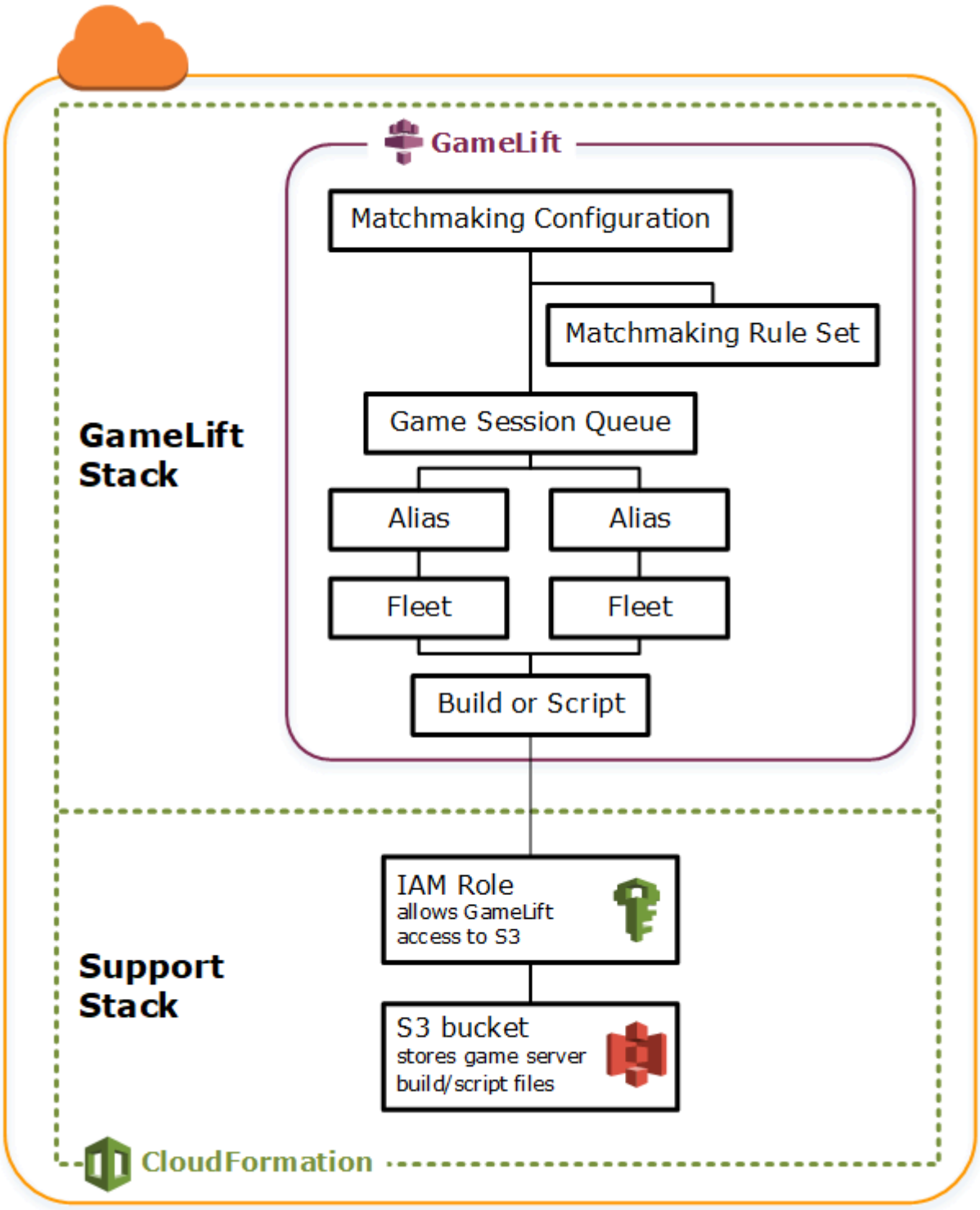
Pile per una singola posizione

Da gestire Amazon GameLift Servers risorse in un'unica posizione, consigliamo una struttura a due stack:

- Support stack: questo stack contiene risorse che Amazon GameLift Servers le risorse dipendono da. Come minimo, questo stack deve includere il bucket S3 in cui si memorizzano i file di script Realtime o il server di gioco personalizzato. Lo stack dovrebbe includere anche un ruolo IAM che fornisca Amazon GameLift Servers autorizzazione a recuperare i file dal bucket S3 durante la creazione di un Amazon GameLift Servers risorsa di compilazione o script. Questo stack potrebbe contenere anche altre AWS risorse utilizzate con il gioco, come tabelle DynamoDB, cluster Amazon Redshift e funzioni Lambda.
- Amazon GameLift Servers stack: questo stack contiene tutti i tuoi Amazon GameLift Servers risorse, tra cui la build o lo script, un set di flotte, alias e la coda delle sessioni di gioco. AWS

CloudFormation crea una risorsa di compilazione o script con file archiviati nella posizione del bucket S3 e distribuisce la build o lo script su una o più risorse del parco risorse. Ogni parco istanze deve avere un alias corrispondente. La coda delle sessioni di gioco fa riferimento ad alcuni o tutti gli alias del parco istanze. Se lo utilizzi FlexMatch per il matchmaking, questo stack contiene anche una configurazione di matchmaking e un set di regole.

Il diagramma seguente illustra una struttura a due livelli per la distribuzione delle risorse in una singola regione. AWS



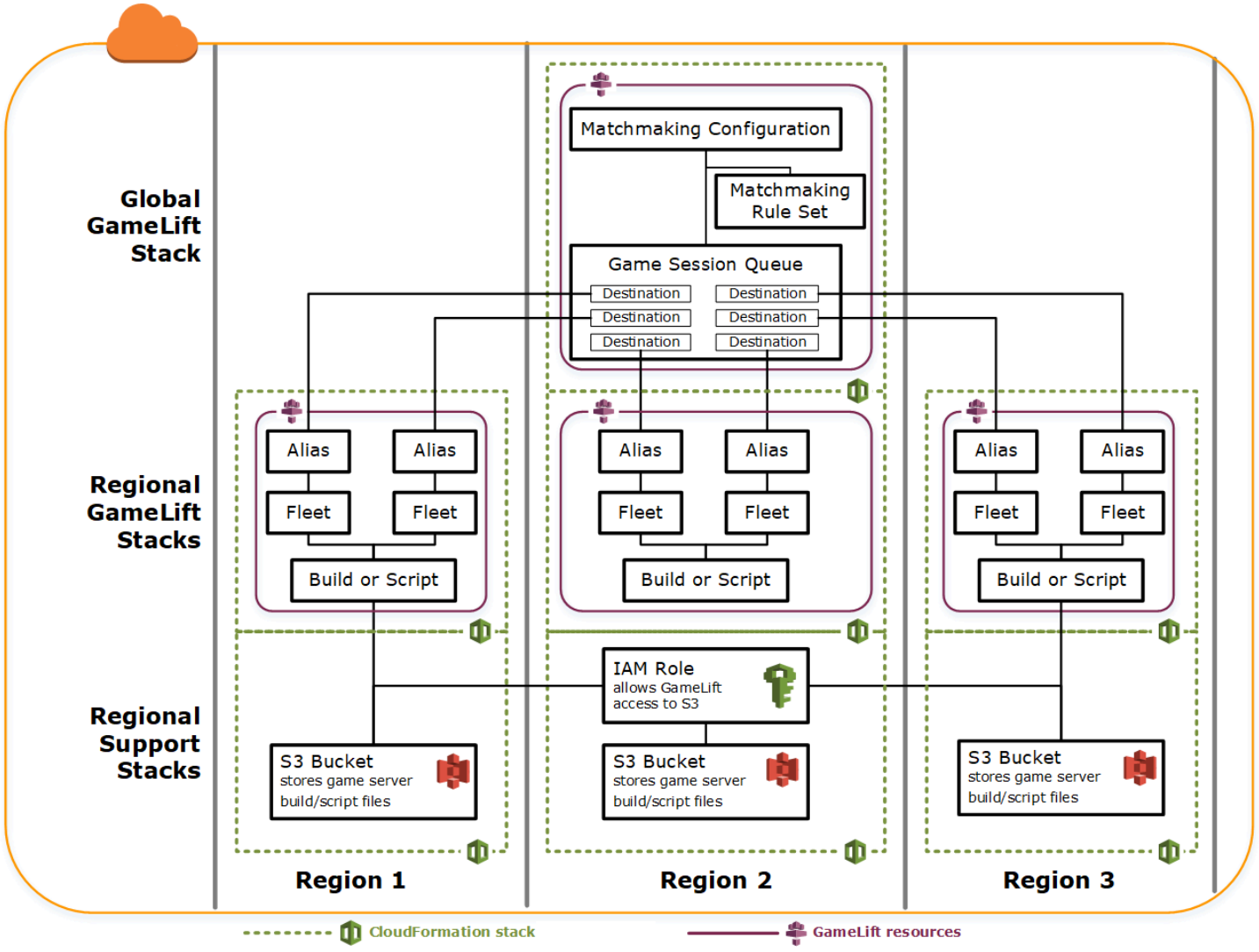
Pile per più regioni

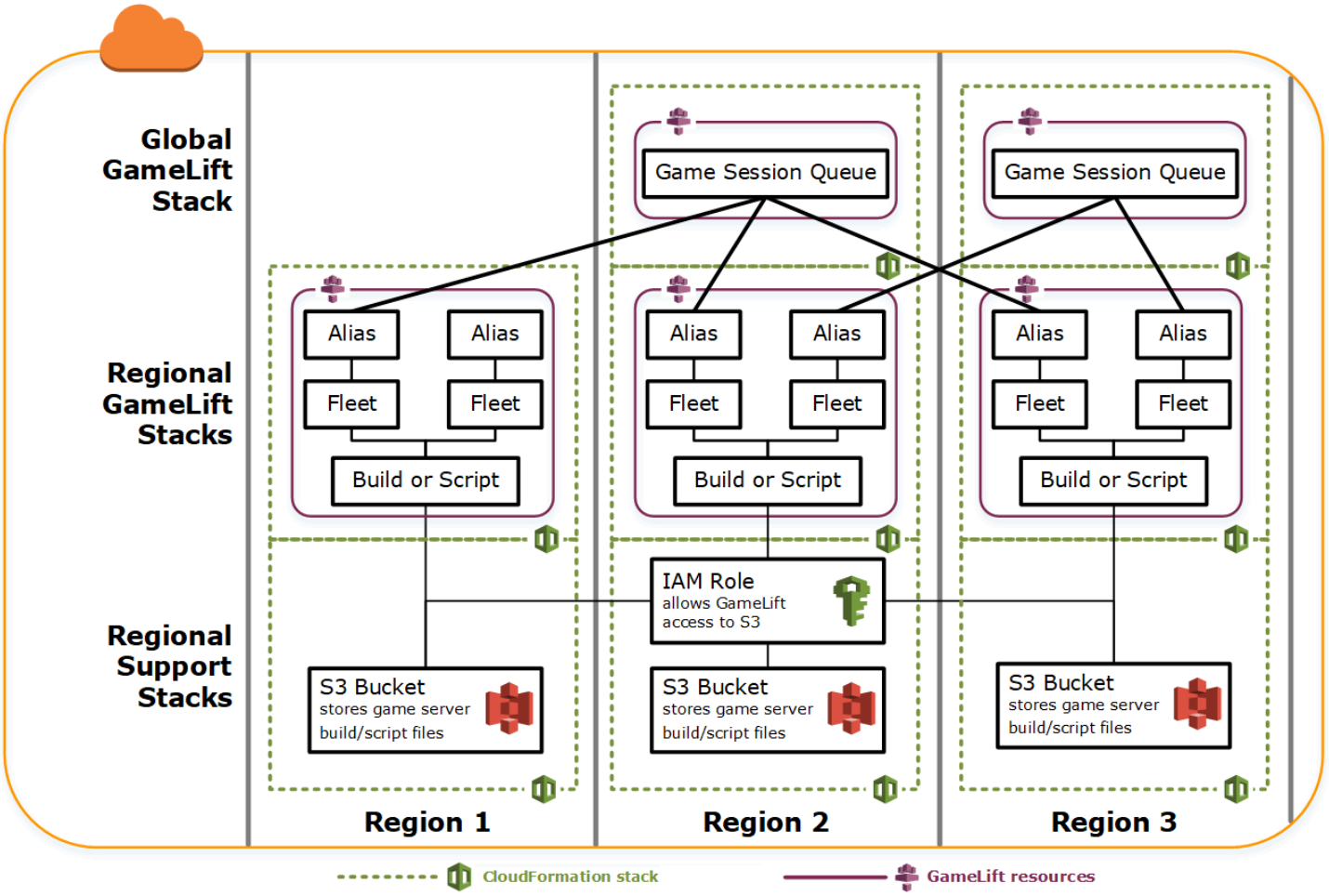
Quando distribuisci il tuo gioco in più di una regione, tieni presente come le risorse possono interagire tra le regioni. Alcune risorse, come Amazon GameLift Servers le flotte possono fare riferimento solo ad altre risorse nella stessa regione. Altre risorse, ad esempio Amazon GameLift Servers

queue, sono indipendenti dalla regione. Da gestire Amazon GameLift Servers risorse in più regioni, consigliamo la seguente struttura.

- Stack di supporto regionali: questi stack contengono risorse che Amazon GameLift Servers le risorse dipendono da. Questo stack deve includere il bucket S3 in cui si memorizzano il server di gioco personalizzato o i file di script Realtime. Potrebbe contenere anche altre AWS risorse per il tuo gioco, come tabelle DynamoDB, cluster Amazon Redshift e funzioni Lambda. Molte di queste risorse sono specifiche per regione, quindi è necessario crearle in ogni regione. Amazon GameLift Servers necessita inoltre di un ruolo IAM che consenta l'accesso a queste risorse di supporto. Poiché un ruolo IAM è indipendente dalla regione, è necessaria una sola risorsa di ruolo, collocata in ogni regione e referenziata in tutti gli altri stack di supporto.
- Regionale Amazon GameLift Servers pile: questa pila contiene Amazon GameLift Servers risorse che devono esistere in ogni regione in cui viene distribuito il gioco, tra cui la build o lo script, un set di flotte e alias. AWS CloudFormation crea una risorsa di compilazione o script con file in una posizione del bucket S3 e distribuisce la build o lo script su una o più risorse del parco risorse. Ogni parco istanze deve avere un alias corrispondente. La coda delle sessioni di gioco fa riferimento ad alcuni o tutti gli alias del parco istanze. Puoi mantenere un modello per descrivere questo tipo di stack e utilizzarlo per creare set di risorse identici in ogni regione.
- Globale Amazon GameLift Servers pila: questo stack contiene la coda della sessione di gioco e le risorse di matchmaking. Queste risorse possono essere localizzate in qualsiasi regione e di solito sono posizionate nella stessa regione. La coda può fare riferimento ai parchi istanze o agli alias che si trovano in qualsiasi regione. Per posizionare code aggiuntive in regioni diverse, crea altri stack globali.

I diagrammi seguenti illustrano una struttura multistack per la distribuzione di risorse in diverse regioni. AWS Il primo diagramma mostra una struttura per una singola coda di sessioni di gioco. Il secondo diagramma mostra una struttura con più code.


Global GameLift Stack
Regional GameLift Stacks
Regional Support Stacks
Matchmaking Configuration
Matchmaking Rule Set
Game Session Queue
Destination
Alias
Fleet
Build or Script
IAM Role
allows GameLift access to S3
S3 Bucket
stores game server build/script files
Region 1
Region 2
Region 3
CloudFormation stack
GameLift resources



Aggiornamento delle build

Amazon GameLift Servers le build sono immutabili, così come la relazione tra una build e una flotta. Di conseguenza, quando aggiorni le risorse di hosting per utilizzare un nuovo set di file della build di gioco, è necessario effettuare quanto segue:

- Creare una nuova build utilizzando il nuovo set di file (sostituzione).
- Creare un nuovo set di parchi istanze per distribuire la nuova build del gioco (sostituzione).
- Reindirizzare gli alias per puntare ai nuovi parchi istanze (aggiornamento senza interruzioni).

Per ulteriori informazioni, consulta Aggiornare i comportamenti delle risorse dello stack nella Guida per l'utente.AWS CloudFormation

Distribuisci automaticamente gli aggiornamenti delle build

Quando si aggiorna uno stack contenente le relative risorse di build, fleet e alias, il AWS CloudFormation comportamento predefinito prevede l'esecuzione automatica di questi passaggi in sequenza. Attivi questo aggiornamento caricando prima i nuovi file della build in un nuovo percorso S3. Quindi modifichi il modello di AWS CloudFormation build in modo che punti alla nuova posizione S3. Quando aggiorni lo stack con il nuovo percorso S3, viene attivata la seguente sequenza AWS CloudFormation :

1. Recupera i nuovi file da S3, li convalida e ne crea uno nuovo Amazon GameLift Servers costruire.
2. Aggiorna il riferimento della build nel modello del parco istanze che attiva la creazione di nuovi parchi istanze.
3. Una volta che i nuovi parchi istanze sono attivi, aggiorna il riferimento del parco istanze nell'alias che attiva l'aggiornamento dell'alias per indirizzare i nuovi parchi istanze.
4. Elimina il vecchio parco istanze.
5. Elimina la vecchia build.

Se la coda delle sessioni di gioco utilizza gli alias del parco istanze, il traffico dei giocatori viene automaticamente trasferito ai nuovi parchi istanze non appena gli alias vengono aggiornati. I vecchi parchi istanze vengono gradualmente svuotati dei giocatori alla fine delle sessioni di gioco. Il dimensionamento automatico gestisce il compito di aggiungere e rimuovere le istanze da ogni gruppo di parchi istanze quando varia il traffico dei giocatori. In alternativa, puoi specificare il numero iniziale di istanze desiderato per aumentare rapidamente lo switch e abilitare il dimensionamento automatico in un secondo momento.

È inoltre possibile impostare la AWS CloudFormation conservazione delle risorse anziché eliminarle. Per ulteriori informazioni, consulta RetainResources nella documentazione di riferimento dell'API AWS CloudFormation .

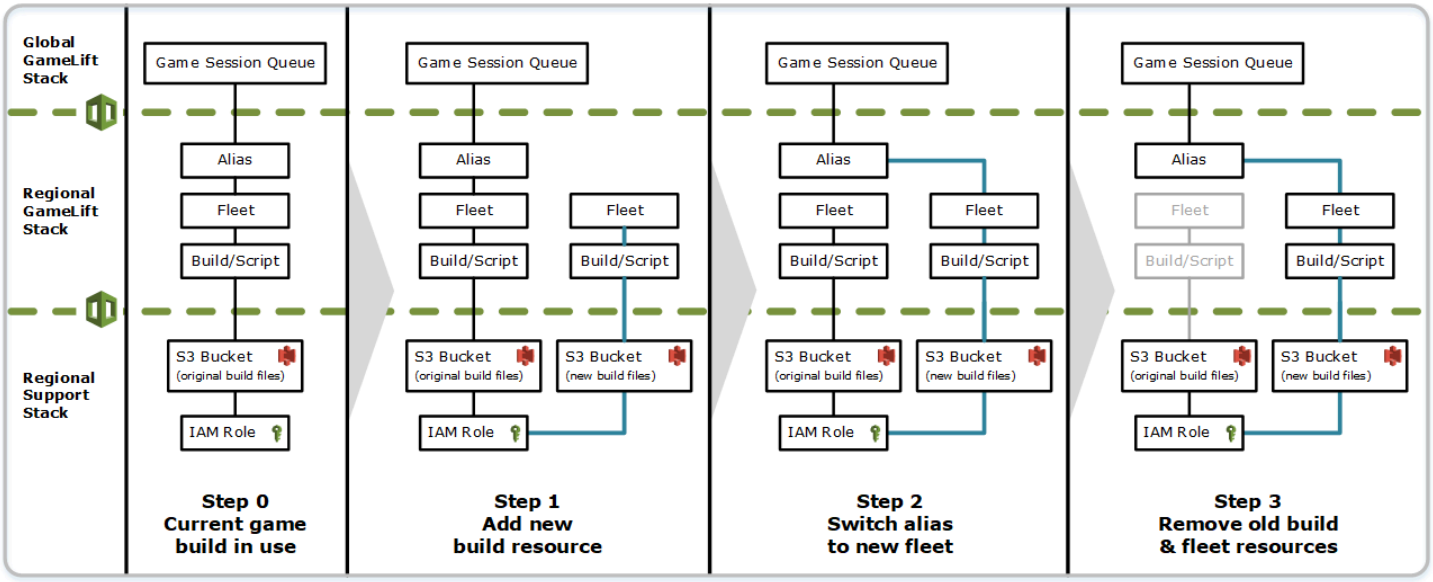
Distribuisci gli aggiornamenti delle build manualmente

Se vuoi avere maggiore controllo su quando i nuovi parchi istanze sono disponibili per i giocatori, hai diverse opzioni. Puoi scegliere di gestire gli alias manualmente utilizzando il Amazon GameLift Servers console o CLI. In alternativa, invece di aggiornare il modello di build per sostituire la build e i parchi istanze, è possibile aggiungere al modello un secondo set di definizioni di build e parco istanze. Quando aggiorni il modello, AWS CloudFormation crea una seconda risorsa di build e le

flotte corrispondenti. Poiché le risorse esistenti non vengono sostituite, non vengono cancellate e gli alias continuano a puntare i parchi instanze originali.

Il vantaggio principale di questo approccio è che ti offre la flessibilità. Puoi creare risorse separate per la nuova versione della tua build, testare le nuove risorse e controllare quando i nuovi parchi istanze diventano disponibili per i giocatori. Un potenziale svantaggio è che richiede il doppio delle risorse in ciascuna regione per un breve periodo di tempo.

Il diagramma seguente illustra tale processo.



Come funzionano i rollback

Quando si esegue un aggiornamento delle risorse, se un passaggio non viene completato, AWS CloudFormation avvia automaticamente un rollback. Questo processo ripristina ogni passaggio della sequenza, eliminando le risorse appena create.

Se è necessario attivare manualmente un rollback, modifica la chiave di percorso S3 del modello di build sul percorso originale e aggiorna lo stack. Una nuova Amazon GameLift Servers build e fleet vengono creati e l'alias passa alla nuova flotta dopo che la flotta è attiva. Se gestisci gli alias separatamente, devi modificarli per puntare ai nuovi parchi istanze.

Per ulteriori informazioni su come gestire un rollback che fallisce o si blocca, consulta Continuare a ripristinare un aggiornamento nella Guida per l'AWS CloudFormation utente.

Monitoraggio Amazon GameLift Servers

Se stai usando Amazon GameLift Servers FleetIQ come funzionalità autonoma di Amazon EC2, consulta la sezione Security in Amazon EC2 nella Amazon EC2 User Guide.

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon GameLift Servers e le altre tue AWS soluzioni. Esistono tre usi principali delle metriche con Amazon GameLift Servers: per monitorare lo stato del sistema e impostare allarmi, monitorare le prestazioni e l'utilizzo dei server di gioco e gestire la capacità utilizzando l'auto-scaling o manuale.

AWS fornisce i seguenti strumenti di monitoraggio da monitorare Amazon GameLift Servers, segnala quando qualcosa non va e intraprendi azioni automatiche se necessario:

- Amazon GameLift Servers console: utilizza l'interfaccia grafica per gestire il Amazon GameLift Servers risorse e monitora l'attività di hosting dei giochi.
- Amazon CloudWatch: puoi monitorare Amazon GameLift Servers metriche in tempo reale, nonché metriche per altre AWS risorse e applicazioni in esecuzione sui AWS servizi. CloudWatch offre una suite di funzionalità di monitoraggio, tra cui strumenti per creare dashboard personalizzati e la possibilità di impostare allarmi che avvisano o intervengono quando una metrica raggiunge una soglia specificata.
- AWS CloudTrail— acquisisce tutte le chiamate API e gli eventi correlati effettuati da o per conto dell'account per AWS Amazon GameLift Servers e altri AWS servizi. I dati vengono forniti come file di log a un bucket Amazon S3 specificato dall'utente. Puoi identificare quali utenti e account hanno effettuato le chiamate AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate.
- Registri delle sessioni di gioco: puoi inviare messaggi server personalizzati per le tue sessioni di gioco in file di registro archiviati in Amazon S3.

Argomenti

Tieni traccia dell'hosting di giochi in Amazon GameLift Servers console

Usa il Amazon GameLift Servers console per visualizzare e gestire le risorse di hosting dei giochi e le attività di hosting in corso quasi in tempo reale. La console offre un'interfaccia grafica per la maggior parte delle funzionalità dell'API di servizio per Amazon GameLift Servers. Puoi usare la console per:

- Usa la dashboard per un'istantanea di alto livello. Puoi vedere i numeri e lo stato attuale di tutti i tuoi Amazon GameLift Servers risorse di hosting e segui i link per ottenere dettagli sulle singole risorse.
- Gestisci le singole risorse di hosting. Puoi creare, visualizzare ed eliminare tutto Amazon GameLift Servers risorse e aggiornarne le proprietà mutabili. Puoi anche visualizzare determinati tipi di attività di hosting, come eventi e metriche delle prestazioni.
- Interagisci con le attività di gioco e di sessione dei giocatori. Puoi tenere traccia delle sessioni di gioco e delle sessioni dei giocatori per flotta e utilizzare queste informazioni per risolvere i problemi relativi alle sessioni di gioco. Visualizza i dettagli sulle sessioni di gioco, visualizza le sessioni dei giocatori per ogni sessione di gioco e cerca l'attività dei giocatori in più sessioni di gioco. Se necessario, puoi anche chiudere le singole sessioni di gioco.

Argomenti

Dashboard di hosting nel Amazon GameLift Servers pannello di controllo

Usa il Amazon GameLift Servers dashboard della console per ottenere un'istantanea di alto livello sullo stato corrente di Amazon GameLift Servers hosting di risorse nel tuo account. AWS Il Amazon GameLift Servers la dashboard offre una visualizzazione di quanto segue:

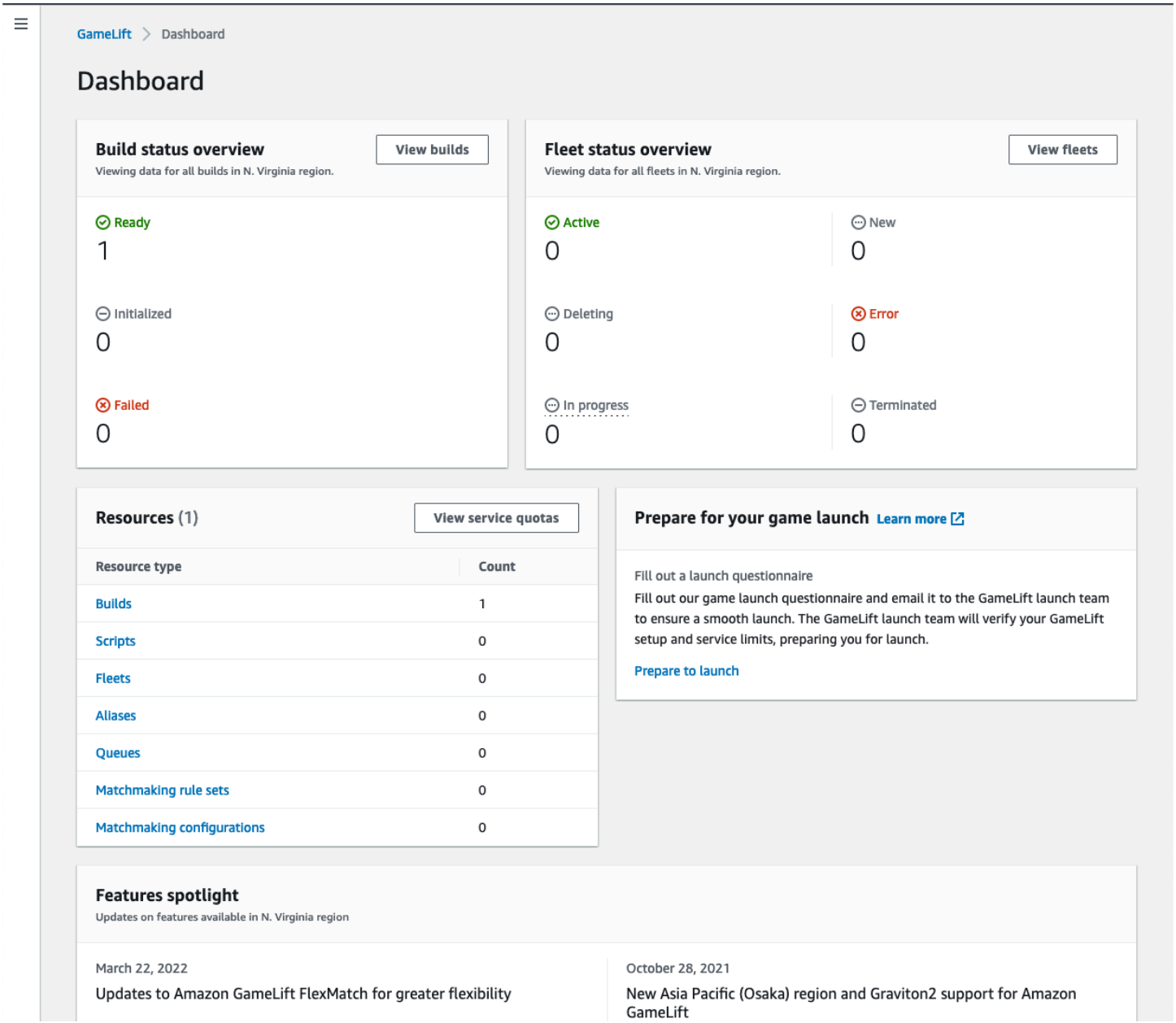
- Il numero di build negli stati Ready, Initialized e Failed. Scegli Visualizza le build per i dettagli sulle build nella tua regione attuale.
- Il numero di flotte in tutti gli stati. Scegli Visualizza flotte per informazioni dettagliate sulle flotte nella tua regione attuale.
- Le tue risorse attuali.
- Annunci di nuove funzionalità e servizi.

Per aprire il Amazon GameLift Servers pannello di controllo

- Nella Amazon GameLift Servers console, nel riquadro di navigazione, scegli Dashboard.

Dalla dashboard, puoi:

- Prepara il gioco per il lancio selezionando Prepara al lancio e compilando il questionario di lancio corrispondente.
- Richiedi aumenti delle quote di servizio in preparazione al lancio o in risposta ai lanci scegliendo Visualizza le quote di servizio.
- Visualizza i post del blog e le informazioni dettagliate sulle nuove funzionalità selezionando il link nella sezione Funzionalità.



Il server di gioco è integrato in Amazon GameLift Servers pannello di controllo

Nella pagina Builds di Amazon GameLift Servers console, puoi visualizzare e gestire tutte le build dei server di gioco su cui hai caricato Amazon GameLift Servers per l'implementazione su EC2 flotte gestite. Nel pannello di navigazione, scegli Hosting, Managed EC2, Builds.

La pagina Builds mostra le seguenti informazioni per ogni build. Puoi modificare il contenuto della tabella secondo necessità utilizzando lo strumento Preferenze (vedi

l'  icona nell'angolo in alto a destra della tabella). Le preferenze personalizzate vengono salvate nell'utente AWS del tuo account e vengono applicate automaticamente ogni volta che visualizzi questa pagina.

> **Note**
>
> La pagina Builds mostra solo le build nella AWS regione corrente.

- Nome: il nome associato alla build caricata.
- Stato: lo stato della build. Visualizza uno dei tre messaggi di stato:
  - Inizializzato: il caricamento non è iniziato o è ancora in corso.
  - Pronto: la build è pronta per la creazione della flotta.
  - Fallita: la compilazione era scaduta prima Amazon GameLift Servers ha ricevuto i file binari.
- Ora di creazione: la data e l'ora in cui hai caricato la build Amazon GameLift Servers.
- ID build: l'ID univoco assegnato alla build al momento del caricamento.
- Versione: l'etichetta della versione associata alla build caricata.
- Sistema operativo: il sistema operativo su cui viene eseguita la build. Il sistema operativo di compilazione determina il sistema operativo Amazon GameLift Servers si installa sulle istanze di un parco istanze.
- Dimensione: la dimensione, in megabyte (MB), del file di build caricato su Amazon GameLift Servers.
- Flotte: il numero di flotte dispiegate durante la build.

Da questa pagina è possibile eseguire queste operazioni:

- Visualizzare i dettagli delle build. Scegli il nome di una build per aprire la relativa pagina dei dettagli della build.
- Creare un nuovo parco istanze da una build. Seleziona una build, quindi scegli Crea flotta.
- Filtrare e ordinare l'elenco delle build. Utilizzare i controlli nella parte superiore della tabella.
- Eliminare una build. Seleziona una build, quindi scegli Elimina.

Dettagli della build

Nella pagina Builds, scegli il nome di una build per aprirne la pagina dei dettagli. La sezione Panoramica della pagina dei dettagli mostra le stesse informazioni di riepilogo della build della pagina Builds. La sezione Flotte mostra un elenco di flotte create con la build, incluse le stesse informazioni di riepilogo della pagina Flotte.

Amazon GameLift ServersRealtimescript nella console Amazon GameLift Servers

Nella pagina Script della Amazon GameLift Serversconsole, puoi visualizzare e gestire tutti gli Amazon GameLift Servers Realtime script caricati Amazon GameLift Servers per la distribuzione su flotte gestite. EC2 Nel pannello di navigazione, scegli Hosting, Scripts.

La pagina Script mostra le seguenti informazioni per ogni script. È possibile modificare il contenuto della tabella in base alle esigenze utilizzando lo strumento Preferenze (vedere l'  icona nell'angolo in alto a destra della tabella). Le preferenze personalizzate vengono salvate nell'utente AWS del tuo account e vengono applicate automaticamente ogni volta che visualizzi questa pagina.

> **Note**
>
> La pagina Script mostra solo gli script nella AWS regione corrente.

- Nome: il nome associato allo script caricato.
- ID: l'ID univoco assegnato allo script durante il caricamento.
- Versione: l'etichetta della versione associata allo script caricato.
- Dimensione: la dimensione, in megabyte (MB), del file di script caricato su. Amazon GameLift Servers
- Ora di creazione: la data e l'ora in cui hai caricato lo script. Amazon GameLift Servers
- Flotte: il numero di flotte dispiegate con lo script.

Da questa pagina è possibile eseguire queste operazioni:

- Visualizza i dettagli dello script. Scegli il nome di una build per aprire la pagina dei dettagli dello script.

- Crea una nuova flotta da uno script. Seleziona uno script, quindi scegli Crea flotta.
- Filtra e ordina l'elenco degli script. Utilizzare i controlli nella parte superiore della tabella.
- Eliminare uno script. Seleziona uno script, quindi scegli Elimina.

Dettagli dello script

Nella pagina Script, scegli il nome di uno script per aprirne la pagina dei dettagli. La sezione Panoramica della pagina dei dettagli mostra le stesse informazioni di riepilogo dello script della pagina Builds. La sezione Flotte mostra un elenco di flotte create con lo script, incluse le stesse informazioni di riepilogo della pagina Flotte.

Flotte nel Amazon GameLift Servers console

Puoi visualizzare le informazioni su tutte le flotte create per ospitare i tuoi giochi Amazon GameLift Servers sotto il tuo AWS account. L'elenco mostra le flotte create nella tua regione attuale. Dalla pagina Flotte, puoi creare una nuova flotta o visualizzare ulteriori dettagli su una flotta. La pagina dei dettagli di una flotta contiene informazioni sull'utilizzo, metriche, dati sulle sessioni di gioco e dati sulle sessioni dei giocatori. Puoi anche modificare un record della flotta o eliminare una flotta.

Per visualizzare la pagina Flotte, scegli Flotte dal pannello di navigazione.

La pagina Parchi istanze per impostazione predefinita contiene le seguenti informazioni riepilogative. Puoi modificare il contenuto della tabella secondo necessità utilizzando lo strumento Preferenze (vedi l'  icona nell'angolo in alto a destra della tabella). Le preferenze personalizzate vengono salvate nell'utente AWS del tuo account e vengono applicate automaticamente ogni volta che visualizzi questa pagina.

- Nome: nome descrittivo dato alla flotta.
- Stato: lo stato della flotta, che può essere uno dei seguenti stati: Nuovo, In download, In costruzione e Attivo.
- Ora di creazione: data e ora di creazione della flotta.
- Tipo di calcolo: il tipo di calcolo utilizzato per ospitare i tuoi giochi. Una flotta può essere una flotta gestita o una EC2 flotta Anywhere.
- Tipo di istanza: il tipo di EC2 istanza Amazon, che determina la capacità di calcolo delle istanze del parco istanze.
- Istanze attive: il numero di EC2 istanze in uso per il parco istanze.

- Istanze desiderate: il numero di EC2 istanze da mantenere attive.
- Sessioni di gioco: il numero di sessioni di gioco attive in esecuzione nella flotta. I dati vengono ritardati di cinque minuti.

Dettagli della flotta nel Amazon GameLift Servers console

Accedi a una pagina dei dettagli della flotta dalla dashboard o dalla pagina Flotte scegliendo il nome della flotta.

Nella pagina dei dettagli della flotta puoi intraprendere le seguenti azioni:

- Aggiorna gli attributi, le impostazioni delle porte e la configurazione di runtime di una flotta.
- Aggiungi o rimuovi le sedi del parco veicoli.
- Modificare le impostazioni della capacità del parco istanze.
- Imposta o modifica l'auto-scaling del target-tracking.
- Eliminare un parco istanze.

Informazioni

Impostazioni della flotta

- Fleet ID: identificativo univoco assegnato alla flotta.
- Nome: il nome della flotta.
- ARN: l'identificatore assegnato a questa flotta. L'ARN di una flotta la identifica come Amazon GameLift Servers risorsa e specifica la regione e l'account. AWS
- Descrizione: una breve descrizione identificabile della flotta.
- Stato: stato attuale della flotta, che può essere Nuovo, In download, In costruzione e Attivo.
- Ora di creazione: data e ora di creazione della flotta.
- Ora di cessazione: la data e l'ora in cui la flotta è stata terminata. Questo campo è vuoto se la flotta è ancora attiva.
- Tipo di parco veicoli: indica se il parco macchine utilizza istanze on-demand o spot.
- EC2 tipo: tipo di EC2 istanza Amazon selezionato per la flotta al momento della creazione.
- Ruolo dell'istanza: un ruolo AWS IAM che gestisce l'accesso alle altre AWS risorse, se ne è stato fornito uno durante la creazione della flotta.

- Certificato TLS: indica se la flotta è abilitata o disabilitata all'uso di un certificato TLS per l'autenticazione di un server di gioco e la crittografia di tutte le comunicazioni client/server.
- Gruppo metrico: il gruppo utilizzato per aggregare le metriche per più flotte.
- Politica di protezione Game Scaling: impostazione attuale per la protezione delle sessioni di gioco per la flotta.
- Numero massimo di sessioni di gioco per giocatore: il numero massimo di sessioni che un giocatore può creare durante il periodo di validità della Politica.
- Periodo di validità: quanto tempo attendere prima di reimpostare il numero di sessioni create da un giocatore.

Dettagli della costruzione

La sezione Dettagli di costruzione mostra la build ospitata nella flotta. Seleziona il nome della build per visualizzare la pagina di dettaglio completa della build.

Configurazione del runtime

La sezione di configurazione di runtime mostra i processi del server da avviare su ciascuna istanza. Include il percorso dei file eseguibili del server di gioco e i parametri opzionali di avvio.

Attivazione della sessione di gioco

La sezione Attivazione della sessione di gioco mostra il numero di processi del server che vengono avviati contemporaneamente e per quanto tempo attendere l'attivazione del processo prima di terminarlo.

EC2 impostazioni delle porte

La sezione Porte mostra le autorizzazioni di connessione della flotta, inclusi gli indirizzi IP e gli intervalli di impostazione delle porte.

Metriche

Nella scheda Parametri è visualizzata una rappresentazione grafica dei parametri del parco istanze nel tempo. Per ulteriori informazioni sull'utilizzo delle metriche in Amazon GameLift Servers, consulta Monitora Amazon GameLift Servers con Amazon CloudWatch.

Eventi

La scheda Events (Eventi) fornisce un log di tutti gli eventi che si sono verificati nel parco istanze, incluso il codice dell'evento, il messaggio e il timestamp. Vedi le descrizioni degli eventi nel Amazon GameLift Servers Riferimento API.

Dimensionamento

La scheda Scaling contiene informazioni sulla capacità della flotta, incluso lo stato attuale e le variazioni di capacità nel tempo. Fornisce inoltre strumenti per aggiornare i limiti di capacità e gestire il dimensionamento automatico.

Capacità di scalabilità

Visualizza le impostazioni correnti della capacità del parco veicoli per ogni sede del parco veicoli. Per ulteriori informazioni sulla modifica dei limiti e della capacità, consultaScalabilità della capacità di hosting di giochi con Amazon GameLift Servers.

- AWS Ubicazione: nome di una sede in cui vengono distribuite le istanze della flotta.
- Stato: stato di hosting dell'ubicazione della flotta. Lo stato della sede deve `ACTIVE` essere tale per poter ospitare le partite.
- Dimensione minima: il numero minimo di istanze che devono essere distribuite nella posizione.
- Istanze desiderate: il numero target di istanze attive per mantenere la posizione. Quando le istanze attive e le istanze desiderate non coincidono, viene avviato un evento di scalabilità per avviare o chiudere le istanze secondo necessità fino a quando le istanze attive non raggiungono le istanze desiderate.
- Dimensione massima: il maggior numero di istanze che possono essere distribuite nella posizione.
- Disponibile: il limite di servizio per le istanze meno il numero di istanze in uso. Questo valore indica il numero massimo di istanze che è possibile aggiungere alla posizione.

Policy di dimensionamento automatico

Questa sezione contiene informazioni sulle politiche di auto-scaling applicate alla flotta. Puoi configurare o aggiornare una politica basata sugli obiettivi. Le politiche basate su regole della flotta, che devono essere definite utilizzando l' AWS SDK o la CLI, vengono visualizzate qui. Per ulteriori informazioni sul dimensionamento, consulta Scalabilità automatica della capacità della flotta con Amazon GameLift Servers.

Cronologia della scalabilità

Visualizza i grafici delle variazioni di capacità nel tempo.

Posizioni

La scheda Sedi elenca tutte le sedi in cui vengono distribuite le istanze del parco istanze. Le sedi includono la regione di origine della flotta e tutte le postazioni remote che sono state aggiunte. Puoi aggiungere o rimuovere sedi direttamente in questa scheda.

- Ubicazione: nome di una sede in cui vengono distribuite le istanze della flotta.
- Stato: stato di hosting dell'ubicazione della flotta. Lo stato della posizione tiene traccia del processo di attivazione delle prime istanze nella posizione. Lo stato della posizione deve essere tale `ACTIVE` per poter ospitare le partite.
- Istanze attive: il numero di istanze con processi server in esecuzione nell'area del parco istanze.
- Server attivi: il numero di processi dei server di gioco in grado di ospitare sessioni di gioco nella sede del parco veicoli.
- Sessioni di gioco: il numero di sessioni di gioco attive sulle istanze presenti nella sede della flotta.
- Sessioni dei giocatori: il numero di sessioni dei giocatori, che rappresentano singoli giocatori, che partecipano a sessioni di gioco attive nella sede della flotta.

Sessioni di gioco

La scheda Game sessions (Sessioni di gioco) mostra le sessioni di gioco passate e presenti ospitate nel parco istanze, oltre ad alcune informazioni dettagliate. Scegli un ID di sessione di gioco per accedere a informazioni aggiuntive sulla sessione di gioco, incluse le sessioni dei giocatori. Per ulteriori informazioni sulle sessioni dei giocatori, consultaSessioni di gioco e di gioco nella Amazon GameLift Servers console.

Sessioni di gioco e di gioco nella Amazon GameLift Servers console

Puoi utilizzare la Amazon GameLift Servers console per lavorare con sessioni di gioco e sessioni di gioco. Per ulteriori informazioni sulle sessioni di gioco e sulle sessioni giocatore, consulta Come i giocatori si connettono ai giochi. La Amazon GameLift Servers console fornisce informazioni e strumenti per aiutarti a risolvere i problemi relativi alle sessioni di gioco.

Cosa puoi fare:

- Esplora le attività delle sessioni di gioco e delle sessioni dei giocatori ospitate su una flotta specifica.
- Cerca le attività delle sessioni di gioco per un giocatore specifico su più flotte.
- Chiudi una sessione di gioco specifica.

Visualizza i dettagli della sessione di gioco

I dati della sessione di gioco e della sessione dei giocatori sono organizzati dalla flotta che ospita la sessione di gioco.

Per accedere alle informazioni sulla sessione di gioco e sulla sessione del giocatore

1. Nella Amazon GameLift Serversconsole, apri il riquadro di navigazione a sinistra. Seleziona un tipo di soluzione di hosting e apri la pagina Fleets. Per esempio:
   - Hosting di flotte ovunque
   - Flotte di hosting EC2, gestite
   - Hosting, contenitori gestiti, flotte
2. Ogni pagina Flotte mostra l'elenco delle flotte attualmente selezionate. Regione AWS Scegli la flotta di cui desideri visualizzare i dati della sessione di gioco.
3. Nella pagina dei dettagli della flotta, apri la scheda Sessioni di gioco. Questa scheda elenca tutte le sessioni di gioco ospitate sulla flotta, insieme a informazioni di riepilogo. Puoi modificare il contenuto della tabella secondo necessità utilizzando lo strumento Preferenze (vedi l'  icona nell'angolo in alto a destra della tabella). Le preferenze personalizzate vengono salvate nell'utente AWS del tuo account e vengono applicate automaticamente ogni volta che visualizzi questa pagina.
4. Scegli una sessione di gioco dall'elenco per visualizzare informazioni aggiuntive.
5. Se la sessione di gioco include i dati delle sessioni dei giocatori, scegli Visualizza le sessioni dei giocatori per aprire lo strumento di ricerca delle sessioni dei giocatori con l'ID della sessione di gioco inserito automaticamente.

I dettagli delle sessioni di gioco includono le seguenti informazioni:

- Stato: stato della sessione di gioco.

  - Attivazione: l'istanza sta avviando una sessione di gioco.
  - Attiva: una sessione di gioco è in corso ed è disponibile per ricevere giocatori, a seconda della politica di creazione dei giocatori della sessione.
  - Terminata: la sessione di gioco è terminata.
- ARN: il nome Amazon Resource della sessione di gioco.
- Nome: nome generato per la sessione di gioco.
- Luogo: il luogo in cui è Amazon GameLift Servers stata ospitata la sessione di gioco.
- Ora di creazione: data e ora di Amazon GameLift Servers creazione della sessione di streaming.
- Ora di fine: data e ora in cui è terminata la sessione di gioco.
- Nome DNS: il nome host della sessione di gioco.
- Indirizzo IP: indirizzo IP specificato per la sessione di gioco.
- Porta: numero di porta utilizzato per connettersi alla sessione di gioco.
- Creator ID: un identificatore univoco del giocatore che ha avviato la sessione di gioco.
- Politica di creazione della sessione dei giocatori: indica se la sessione di gioco accetta nuovi giocatori.
- Politica di protezione della scalabilità del gioco: il tipo di protezione della sessione di gioco da impostare su tutte le nuove istanze che entrano a far Amazon GameLift Servers parte della flotta.

Dati di gioco

Dati sulle proprietà del gioco, formattati come stringa, da inviare alla sessione di gioco all'inizio.

Proprietà del gioco

Dati sulle proprietà del gioco, formattati come key/value coppie, da inviare alla sessione di gioco all'inizio.

Dati di matchmaking

Se la sessione di gioco è stata creata conFlexMatch, i dati di matchmaking descrivono le informazioni sulla configurazione del matchmaking e sul set di regole. Ciò include gli attributi dei giocatori e gli incarichi di squadra di ogni partita. I dati sono in formato JSON. Per ulteriori informazioni sul FlexMatch matchmaking, vedi Build a matchmaker.

Cerca i dati della sessione dei giocatori

Se la tua soluzione di hosting di giochi utilizza sessioni di gioco e offre giocatori unici IDs, puoi esplorare le attività specifiche del giocatore per le sessioni di gioco passate o presenti su più flotte. Apri lo strumento di ricerca delle sessioni Player utilizzando uno dei seguenti metodi:

- Nella Amazon GameLift Servers console, apri il riquadro di navigazione a sinistra e scegli Player session lookup e seleziona il tipo di filtro da utilizzare.
- Quando visualizzi i dettagli della sessione di gioco di una flotta, scegli Visualizza le sessioni dei giocatori. Lo strumento di ricerca si apre con la sessione di gioco con il filtro ID della sessione di gioco preselezionato e il valore della sessione di gioco inserito.

Quando utilizzi lo strumento di ricerca, puoi fornire le seguenti informazioni:

- Un ID di sessione del giocatore per ottenere informazioni su una sessione di gioco specifica.
- Un ID di sessione di gioco per ottenere informazioni su tutte le sessioni dei giocatori per la sessione di gioco richiesta. I risultati rappresentano tutti i giocatori che hanno prenotato uno slot o si sono collegati alla sessione di gioco. Puoi facoltativamente filtrare i risultati in base allo stato della sessione del giocatore.
- Un ID giocatore per ottenere informazioni su tutte le sessioni del giocatore richiesto. I risultati rappresentano tutte le sessioni di gioco a cui il giocatore ha partecipato.

> **Note**
>
> Lo strumento di ricerca cerca tutte le attività delle sessioni dei giocatori tra quelle attualmente selezionate. Regione AWS Se hai più flotte nella Regione, i risultati includono l'attività delle sessioni dei giocatori in tutte le flotte. Per le flotte con più sedi, i risultati includono anche l'attività delle sessioni dei giocatori nelle località remote delle flotte.

Per ciascuna sessione di gioco vengono raccolti i seguenti dati delle sessioni giocatore:

- ID della sessione del giocatore: l'identificatore assegnato alla sessione del giocatore.
- ID giocatore: un identificatore univoco per il giocatore. Scegli questo ID per ottenere ulteriori informazioni sul giocatore.
- ID della sessione di gioco: l'identificatore assegnato alla sessione di gioco.

- Fleet ID: l'identificativo assegnato alla flotta che ha ospitato la sessione di gioco.
- Stato: lo stato della sessione del giocatore. Gli stati possibili sono i seguenti:
  - Riservata: la sessione del giocatore è stata riservata, ma i giocatori non sono connessi.
  - Attiva: la sessione del giocatore è connessa al server di gioco.
  - Completata: la sessione del giocatore è terminata; il giocatore non è più connesso.
  - Timeout: il giocatore non è riuscito a connettersi.
- Ora di creazione: l'ora in cui il giocatore si è connesso alla sessione di gioco.
- Ora di fine: l'ora in cui il giocatore si è disconnesso dalla sessione di gioco.
- Dati di connessione: indirizzo IP, nome DNS e porta utilizzati dal giocatore per connettersi alla sessione di gioco.
- Dati del giocatore: informazioni sul giocatore fornite durante la creazione della sessione di gioco.

Chiudi una sessione di gioco

Usa la Amazon GameLift Servers console per chiudere una sessione di gioco specifica. Questa funzione offre un metodo semplice e veloce per localizzare una sessione di gioco e inviare un segnale per terminarla. Un altro metodo di terminazione richiede di trovare l'istanza fleet su cui è in corso la sessione di gioco, accedere in remoto all'istanza e chiudere manualmente la sessione di gioco.

Puoi interrompere una sessione di gioco per qualsiasi motivo. Il motivo più comune è risolvere una sessione di gioco che non si chiude naturalmente. Di conseguenza, la risorsa di hosting per la sessione di gioco non può essere liberata per ospitare una nuova sessione di gioco e la capacità di hosting della flotta viene ridotta.

> **Note**
>
> Questa funzionalità si basa su determinate impostazioni di configurazione per la tua soluzione di hosting. Presenta le seguenti limitazioni:
>
> - La sessione di gioco deve essere ospitata su una flotta che esegue un server di gioco costruito con server SDK per Amazon GameLift Servers v5 o versioni successive. Se i tuoi server di gioco utilizzano una versione precedente, devi utilizzare l'accesso remoto per eliminare la sessione di gioco.

- Se la sessione di gioco è ospitata su una flotta Anywhere, la flotta deve utilizzare l'Amazon GameLift Serversagente per gestire i processi del server di gioco.

Per terminare una sessione di gioco

1. Nella Amazon GameLift Serversconsole, apri il riquadro di navigazione a sinistra. Seleziona un tipo di soluzione di hosting e apri la pagina Fleets. Per esempio:
   - Hosting di flotte ovunque
   - Flotte di hosting EC2, gestite
   - Hosting, contenitori gestiti, flotte
2. Ogni pagina Flotte mostra l'elenco delle flotte attualmente selezionate. Regione AWS Scegli la flotta che ospita la sessione di gioco che desideri interrompere.
3. Nella pagina dei dettagli della flotta, apri la scheda Sessioni di gioco. Nell'elenco delle sessioni di gioco, seleziona quella che desideri interrompere e scegli il pulsante Termina.
4. Nella sessione di gioco Termina? nella finestra, verifica di chiudere la sessione di gioco corretta e scegli un metodo di terminazione.
   - Chiusura normale della sessione di gioco: questa opzione invia un segnale di chiusura al processo del server che ospita la sessione di gioco. Se la build del server di gioco è stata integrata correttamente perAmazon GameLift Servers, il processo server avvia la sequenza di chiusura della sessione di gioco, notifica la fine della sessione e si Amazon GameLift Servers interrompe. A seconda del design del gioco, la sequenza di spegnimento potrebbe includere passaggi per completare correttamente la sessione di gioco, come il salvataggio dei dati e la notifica ai giocatori attivi. Questo metodo potrebbe richiedere un piccolo ritardo per completare la sequenza di chiusura della sessione di gioco.
   - Chiusura immediata della sessione di gioco: questa opzione invia un segnale a un gestore dei processi per chiudere il processo del server che ospita la sessione di gioco. Questa opzione ignora la normale chiusura della sessione di gioco. È in grado di terminare la sessione di gioco anche quando il processo del server non è in grado di rispondere.
5. Conferma la cessazione della sessione di gioco. Puoi tenere traccia dell'avanzamento della chiusura nella pagina della console delle sessioni di gioco. Lo stato della sessione di gioco cambierà in «Interruzione» e poi in «Interrotta» al termine della sessione di gioco.

Argomenti correlati

- Puoi anche chiudere le sessioni di gioco utilizzando l'SDK e il. AWS AWS CLI Per ulteriori dettagli ed esempi, consulta l'argomento Amazon GameLift Servers API Reference. TerminateGameSession
- Per ulteriori informazioni sull'integrazione dei server di gioco e su come un processo server risponde ai segnali del Amazon GameLift Servers servizio, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

Alias in Amazon GameLift Servers console

La pagina Alias visualizza informazioni su Amazon GameLift Servers alias che indirizzano il traffico verso destinazioni di hosting specifiche. Per gli alias, scegli Hosting, Alias nel pannello di navigazione.

Nella pagina degli alias puoi effettuare le seguenti operazioni:

- Crea un nuovo alias. Scegli Crea alias.
- Filtra e ordina la tabella degli alias. Utilizzare i controlli nella parte superiore della tabella. Puoi modificare il contenuto della tabella secondo necessità utilizzando lo strumento Preferenze (vedi l'  icona nell'angolo in alto a destra della tabella). Le preferenze personalizzate vengono salvate nell'utente AWS del tuo account e vengono applicate automaticamente ogni volta che visualizzi questa pagina.
- Visualizzare i dettagli degli alias. Scegli un nome alias per aprire la pagina dei dettagli dell'alias.
- Eliminare un alias. Scegli un alias, quindi scegli Elimina.

Dettagli dell'alias

La pagina dei dettagli dell'alias mostra informazioni sull'alias.

In questa pagina, è possibile:

- Modifica un alias. Scegli Modifica.
- Visualizza le flotte che hai associato all'alias.
- Eliminare un alias. Scegli Elimina.

Le informazioni dettagliate sugli alias includono:

- ID: il numero univoco utilizzato per identificare l'alias.
- Descrizione: la descrizione dell'alias.
- ARN: il nome Amazon Resource dell'alias.
- Creazione: data e ora di creazione dell'alias.
- Ultimo aggiornamento: la data e l'ora dell'ultimo aggiornamento dell'alias.
- Tipo di routing: il tipo di routing per l'alias, che può essere uno dei seguenti:
  - Semplice: indirizza il traffico dei giocatori verso un ID flotta specificato. È possibile aggiornare l'ID del parco istanze per un alias in qualsiasi momento.
  - Terminale: restituisce un messaggio al client. Ad esempio, puoi indirizzare i giocatori che utilizzano un out-of-date client verso un luogo in cui possono ottenere un upgrade.
- Tag: coppie di chiavi e valori utilizzate per identificare l'alias.

Code per le sessioni di gioco nel Amazon GameLift Servers console

Puoi visualizzare le informazioni su tutte le code, che vengono utilizzate per elaborare le richieste di nuove sessioni di gioco. La pagina delle code mostra le code delle sessioni di gioco nella zona attualmente selezionata. Regione AWS Nella pagina Queues (Code), è possibile creare una nuova coda, eliminare le code esistenti oppure aprire una pagina di dettaglio per una coda selezionata. Ogni pagina dei dettagli della coda contiene i dati relativi alla configurazione e alle metriche della coda. Per ulteriori informazioni sulle code, consulta Gestione del posizionamento delle sessioni di gioco con Amazon GameLift Servers code.

La pagina delle code mostra le seguenti informazioni di riepilogo per ogni coda. È possibile modificare il contenuto della tabella in base alle esigenze utilizzando lo strumento Preferenze (vedere l'  icona nell'angolo in alto a destra della tabella). Le preferenze personalizzate vengono salvate nell'utente AWS del tuo account e vengono applicate automaticamente ogni volta che visualizzi questa pagina.

- Nome della coda: il nome assegnato alla coda. Le richieste di nuove sessioni di gioco specificano una coda utilizzando questo nome.
- Timeout della coda: periodo massimo, in secondi, durante il quale una richiesta di posizionamento della sessione di gioco rimane in coda prima del timeout.
- Destinazioni in coda: numero di flotte elencate nella configurazione della coda. Amazon GameLift Servers inserisce nuove sessioni di gioco su qualsiasi flotta in coda.

Visualizza i dettagli della coda

Puoi accedere a informazioni dettagliate su qualsiasi coda, inclusi la configurazione e i parametri della coda. Per aprire una pagina dei dettagli della coda, vai alla pagina Code e scegli un nome per la coda.

Nella pagina dei dettagli della coda è visualizzata una tabella di riepilogo e schede con informazioni aggiuntive. In questa pagina puoi eseguire le seguenti operazioni:

- Aggiorna la configurazione della coda, l'elenco delle destinazioni e le policy di latenza dei giocatori. Scegli Modifica.
- Eliminare una coda. Dopo aver eliminato una coda, tutte le richieste di nuove sessioni di gioco che fanno riferimento al nome della coda avranno esito negativo. Scegli Elimina.

> **Note**
>
> Per ripristinare una coda eliminata, crea una nuova coda con il nome della coda eliminata.

Informazioni

Panoramica

La sezione Panoramica mostra l'Amazon Resource Name (ARN) della coda e il timeout. È possibile utilizzare l'ARN per fare riferimento alla coda in altre azioni o aree di Amazon GameLift Servers. Il timeout è il periodo di tempo massimo, in secondi, durante il quale una richiesta di posizionamento della sessione di gioco rimane in coda prima del timeout.

Notifiche di eventi

La sezione Notifica degli eventi elenca l'argomento SNS Amazon GameLift Servers pubblica le notifiche degli eventi e i dati sugli eventi che vengono aggiunti a tutti gli eventi creati da questa coda.

Tag

La tabella Tag mostra le chiavi e i valori utilizzati per etichettare la risorsa. Per ulteriori informazioni sull'etichettatura, consulta Etichettatura delle risorse AWS.

Metriche

Nella scheda Metrics (Parametri) è mostrata una rappresentazione grafica dei parametri della coda nel tempo.

Le metriche sulla coda includono una serie di informazioni che descrivono l'attività di posizionamento in coda, compresi i posizionamenti riusciti organizzati per regione. Puoi utilizzare i dati della regione per capire dove stai ospitando i tuoi giochi. Le metriche di posizionamento a livello regionale possono aiutare a individuare problemi relativi alla progettazione generale delle code.

Le metriche della coda sono disponibili anche in Amazon. CloudWatch Per le descrizioni delle metriche disponibili, consulta. Parametri Amazon GameLift Servers per code

Destinazioni

La scheda Destinations (Destinazioni) mostra tutti i parchi istanze o gli alias elencati per la coda.

Quando Amazon GameLift Servers cerca le risorse disponibili nelle destinazioni per ospitare una nuova sessione di gioco, cerca nell'ordine predefinito elencato qui. Finché c'è capacità sulla prima destinazione elencata, Amazon GameLift Servers vi inserisce nuove sessioni di gioco. Puoi fare in modo che le singole richieste di posizionamento delle sessioni di gioco abbiano la priorità sull'ordine predefinito fornendo i dati di latenza dei giocatori. Questi dati dicono Amazon GameLift Servers per cercare una destinazione disponibile con la latenza media dei giocatori più bassa. Per ulteriori informazioni sulla progettazione delle code, consulta. Personalizza una coda di sessioni di gioco

Posizionamento della sessione

Politiche di latenza dei giocatori

La sezione Criteri di latenza di Player mostra tutti i criteri utilizzati dalla coda. Le tabelle elencano le politiche nell'ordine in cui vengono applicate.

Posizioni

La sezione Luoghi mostra i luoghi in cui questa coda può inserire una sessione di gioco.

Priorità

La sezione Priorità mostra l'ordine in cui la coda valuta i dettagli di una sessione di gioco.

Ordine di ubicazione

La sezione Ordine delle posizioni mostra l'ordine predefinito utilizzato dalla coda per piazzare le sessioni di gioco. La coda utilizza questo ordine se non sono stati definiti altri tipi di priorità.

Monitora Amazon GameLift Servers con Amazon CloudWatch

Puoi monitorare Amazon GameLift Servers utilizzando Amazon CloudWatch, un AWS servizio che raccoglie dati grezzi e li elabora in metriche leggibili quasi in tempo reale. Queste statistiche vengono conservate per un periodo di 15 mesi per fornire una prospettiva storica delle prestazioni del server di gioco che ospita Amazon GameLift Servers. Puoi impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Le seguenti tabelle elencano i parametri e le dimensioni di Amazon GameLift Servers. Tutte le metriche disponibili sono disponibili anche nella CloudWatch Amazon GameLift Servers console, che fornisce i dati sotto forma di set di grafici personalizzabili. Per accedere alle CloudWatch metriche dei tuoi giochi, usa l' AWS Management Console AWS CLI, o l'API. CloudWatch

Se una metrica non ha una posizione, utilizza la posizione principale.

Dimensioni per i parametri Amazon GameLift Servers

Amazon GameLift Servers supporta l'applicazione di filtri ai parametri per le seguenti dimensioni.

Dimensione	Descrizione
`Location`	Filtra le metriche per una località di distribuzione del parco veicoli. Se una metrica non ha una posizione, utilizza la posizione principale.
`FleetId`	Filtra i parametri per un singolo parco istanze. Questa dimensione può essere utilizzata con tutti i parametri di parchi istanze per istanze, processi server, sessioni di gioco e sessioni giocatore.
`MetricGroup`	Filtra i parametri per una raccolta di parchi istanze. Aggiungi un parco veicoli a un gruppo di metriche aggiungendo il nome del gruppo di metriche agli attributi del parco veicoli (vedi UpdateFleetAttribu

Dimensione	Descrizione
	tes()). Questa dimensione può essere utilizzata con tutti i parametri di parchi istanze per istanze, processi server, sessioni di gioco e sessioni giocatore.
`QueueName`	Filtra i parametri per una singola coda. Questa dimensione viene utilizzata solo con i parametri per le code delle sessioni di gioco.
`ConfigurationName`	Filtra i parametri per una singola configurazione di abbinamento. Questa dimensione viene utilizzat a solo con i parametri per le configurazioni di abbinamenti.
`ConfigurationName-RuleName`	Filtra i parametri per l'intersezione di una configura zione di abbinamento e una regola di abbinamen to. Questa dimensione viene utilizzata solo con parametri per regole di abbinamento.
`InstanceType`	Filtra le metriche per la designazione del tipo di EC2 istanza, ad esempio «c4.large». Questa dimensione viene utilizzata con parametri per istanze Spot.
`OperatingSystem`	Filtra i parametri per il sistema operativo di un'istanz a. Questa dimensione viene utilizzata con i parametri per le istanze Spot.
`GameServerGroup`	Filtra i parametri FleetIQ per un gruppo di server di gioco.

Parametri di Amazon GameLift Servers per parchi istanze

Il namespace `AWS/GameLift` include i seguenti parametri relativi all'attività in un parco istanze o un gruppo di parchi istanze. I parchi istanze vengono utilizzati con una soluzione Amazon GameLift Servers gestita. Il Amazon GameLift Servers servizio invia metriche a ogni minuto. CloudWatch

Istanze

Parametro	Descrizione
`ActiveInstances`	Istanze con stato ACTIVE (Attivo), ovvero istanze che stanno eseguendo processi server attivi. Il numero include le istanze inattive e quelle che ospitano una o più sessioni di gioco. Questo parametro misura la capacità di istanze totale corrente e può essere utilizzato con la scalabilità automatica. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`DesiredInstances`	Il numero target di istanze attive che Amazon GameLift Servers sta elaborando per mantenerl e nel parco istanze. Con la scalabilità automatica questo valore viene determinato sulla base delle policy di scalabilità attualmente applicate. Senza scalabilità automatica, questo valore viene impostato manualmente. Questo parametro non è disponibile quando si visualizzano i dati per gruppi di parametri del parco istanze. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`IdleInstances`	Istanze attive che ospitano attualmente zero (0) sessioni di gioco. Questo parametro misura la

Parametro	Descrizione
	capacità disponibile ma non utilizzata. e può essere utilizzato con la scalabilità automatica. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`MaxInstances`	Numero massimo di istanze consentite per il parco istanze. Tale numero determina la capacità massima durante la scalabilità automatica o manuale. Questo parametro non è disponibile quando si visualizzano i dati per gruppi di parametri del parco istanze. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`MinInstances`	Numero minimo di istanze consentite per il parco istanze. Tale numero determina la capacità minima durante la scalabilità automatica o manuale. Questo parametro non è disponibile quando si visualizzano i dati per gruppi di parametri del parco istanze. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione

Parametro	Descrizione
`PercentIdleInstances`	Percentuale di tutte le istanze attive che sono inattive (calcolata come `IdleInstances / ActiveInstances`). Questo parametro può essere utilizzato per la scalabilità automatica. Unità: percentuale CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`RecycledInstances`	Numero di istanze spot che sono state riciclate e sostituite. Amazon GameLift Serversricicla le istanze spot che attualmente non ospitano sessioni di gioco e che hanno un'alta probabilità di interruzione. Unità: numero CloudWatch Statistiche pertinenti: somma, media, minima, massima Dimensioni: Ubicazione
`InstanceInterruptions`	Numero di istanze Spot che sono state interrotte. Unità: numero CloudWatch Statistiche pertinenti: somma, media, minima, massima Dimensioni: Ubicazione

Parametro	Descrizione
`CPUUtilization`	EC2 metrico. Perché Amazon GameLift Servers questa metrica rappresenta le prestazioni hardware di tutte le istanze attive in una flotta. La percentua le di tempo fisico della CPU EC2 utilizzata da Amazon per eseguire l'istanza, che include il tempo impiegato per eseguire sia il codice utente che il EC2 codice Amazon. Gli strumenti del tuo sistema operativo possono mostrare una percentuale diversa CloudWatch rispetto a fattori quali la simulazione di dispositivi legacy, la configurazione di dispositivi non legacy, i carichi di lavoro che richiedono interruzioni, la migrazione in tempo reale e l'aggiornamento in tempo reale. Unità: percentuale
`NetworkIn`	EC2 metrico. Perché Amazon GameLift Servers questa metrica rappresenta le prestazioni hardware di tutte le istanze attive in una flotta. Il numero di byte ricevuti dall'istanza su tutte le interfacce di rete. Identifica il volume del traffico di rete in entrata su un'applicazione di una singola istanza. Unità: byte
`NetworkOut`	EC2 metrica. Perché Amazon GameLift Servers questa metrica rappresenta le prestazioni hardware di tutte le istanze attive in una flotta. Il numero di byte inviati dall'istanza su tutte le interfacce di rete. Identifica il volume del traffico di rete in uscita verso un'applicazione di una singola istanza. Unità: byte

Parametro	Descrizione
`DiskReadBytes`	EC2 metrica. Perché Amazon GameLift Servers questa metrica rappresenta le prestazioni hardware di tutte le istanze attive in una flotta. Byte letti da tutti i volumi di instance store disponibili per l'istanza . Questo parametro viene utilizzato per determina re il volume dei dati che l'applicazione legge dal disco rigido dell'istanza. È possibile utilizzarlo per determinare la velocità dell'applicazione. Unità: byte
`DiskWriteBytes`	EC2 metrico. Perché Amazon GameLift Servers questa metrica rappresenta le prestazioni hardware di tutte le istanze attive in una flotta. Byte scritti in tutti i volumi di instance store disponibili per l'istanza . Questo parametro viene utilizzato per determina re il volume dei dati che l'applicazione scrive sul disco rigido dell'istanza. È possibile utilizzarlo per determinare la velocità dell'applicazione. Unità: byte
`DiskReadOps`	EC2 metrico. Perché Amazon GameLift Servers questa metrica rappresenta le prestazioni hardware di tutte le istanze attive in una flotta. Operazioni di lettura completate da tutti i volumi di instance store disponibili per l'istanza in un determinato periodo di tempo. Per calcolare le I/O operazioni medie al secondo (IOPS) per il periodo, dividi il totale delle operazioni nel periodo per il numero di secondi in quel periodo. Unità: numero

Parametro	Descrizione
`DiskWriteOps`	EC2 metrico. Perché Amazon GameLift Servers questa metrica rappresenta le prestazioni hardware di tutte le istanze attive in una flotta. Operazioni di scrittura completate in tutti i volumi di instance store disponibili per l'istanza in un determinato periodo di tempo. Per calcolare le I/O operazioni medie al secondo (IOPS) per il periodo, dividi il totale delle operazioni nel periodo per il numero di secondi in quel periodo. Unità: numero

Processi server

Parametro	Descrizione
`ActiveServerProcesses`	Processi server con stato ACTIVE (Attivo), ovvero processi in esecuzione e in grado di ospitare sessioni di gioco. Il numero include i processi server inattivi e quelli che ospitano una o più sessioni di gioco. Questo parametro misura la capacità di processi server totale corrente. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`HealthyServerProcesses`	Processi server attivi che risultano integri. Questo è un parametro utile per tenere traccia dello stato complessivo dei server di gioco del parco istanze. Unità: numero

Parametro	Descrizione
	CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`PercentHealthyServerProcesses`	Percentuale di tutti i processi server attivi che risultano integri (calcolata come `HealthyServerProcesses / ActiveServerProcesses`). Unità: percentuale CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`ServerProcessAbnormalTerminations`	Processi server che sono stati chiusi a causa di circostanze anomale dall'ultimo rapporto. Questo parametro include le terminazioni avviate dal servizio Amazon GameLift Servers. Ciò si verifica quando un processo del server smette di rispondere, segnala costantemente i controlli di integrità non riusciti o non termina in modo corretto (chiamando () ProcessEnding). Unità: numero CloudWatch Statistiche pertinenti: somma, media, minimo, massimo Dimensioni: Ubicazione

Parametro	Descrizione
`ServerProcessActivations`	Processi server che sono passati correttamente dallo stato ACTIVATING (Attivazione in corso) allo stato ACTIVE (Attivo) dall'ultimo rapporto. I processi server non possono ospitare sessioni di gioco se non sono attivi. Unità: numero CloudWatch Statistiche pertinenti: somma, media, minima, massima Dimensioni: Ubicazione
`ServerProcessTerminations`	Processi server che sono stati chiusi dall'ultimo rapporto. Ciò include tutti i processi server che sono passati allo stato TERMINATED (Chiuso) per qualunque motivo, anche le terminazioni normali e non normali. Unità: numero CloudWatch Statistiche pertinenti: somma, media, minima, massima Dimensioni: Ubicazione

Sessioni di gioco

Parametro	Descrizione
`ActivatingGameSessions`	Sessioni di gioco con stato ACTIVATING (Attivazione in corso), ovvero sessioni in fase di avvio. Le sessioni di gioco possono ospitare i giocatori se non sono attive. Numeri alti per un lungo periodo di tempo possono indicare che le sessioni di gioco non passano dallo stato ACTIVATING (Attivazione

Parametro	Descrizione
	in corso) allo stato ACTIVE (Attivo). e può essere utilizzato con la scalabilità automatica. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`ActiveGameSessions`	Sessioni di gioco con stato ACTIVE (Attivo), ovvero sessioni in grado di ospitare giocatori e che ospitano zero o più giocatori. Questo parametro misura il numero totale di sessioni di gioco attualmente ospitate. e può essere utilizzato con la scalabilità automatica. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione

Parametro	Descrizione
`AvailableGameSessions`	Processi server attivi e integri che non vengono attualmente utilizzati per ospitare una sessione di gioco e che possono avviare una nuova sessione di gioco senza ritardi per avviare nuovi processi o istanze del server. e può essere utilizzato con la scalabilità automatica. **Note** Per le flotte che limitano le attivazioni simultanee di sessioni di gioco, utilizza la metrica. `ConcurrentActivatableGameSessions` Questa metrica rappresenta in modo più accurato il numero di nuove sessioni di gioco che possono iniziare senza alcun tipo di ritardo. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione

Parametro	Descrizione
`ConcurrentActivatableGameSessions`	Processi server attivi e integri che non vengono attualmente utilizzati per ospitare una sessione di gioco e che possono iniziare immediatamente una nuova sessione di gioco. Questa metrica differisce `AvailableGameSessions` nel modo seguente: non conta i processi del server che attualmente non possono attivare una nuova sessione di gioco a causa dei limiti alle attivazioni delle sessioni di gioco. (Vedi l'impostazione RuntimeConfigurationopzionale della flotta). `MaxConcurrentGameSessionActivations` Per le flotte che non limitano le attivazioni delle sessioni di gioco, questa metrica è identica a. `AvailableGameSessions` Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima Dimensioni: Ubicazione
`PercentAvailableGameSessions`	Percentuale delle slot di sessioni di gioco su tutti i processi server attivi (integri o non integri) che non sono attualmente in uso (calcolata come `AvailableGameSessions / [ActiveGameSessions + AvailableGameSessions + unhealthy server processes]`). e può essere utilizzato con la scalabilità automatica. Unità: percentuale CloudWatch Statistiche pertinenti: media Dimensioni: ubicazione

Parametro	Descrizione
`GameSessionInterruptions`	Il numero di sessioni di gioco su istanze Spot che sono state interrotte. Unità: numero CloudWatch Statistiche pertinenti: somma, media, minima, massima Dimensioni: Ubicazione

Sessioni giocatore

Parametro	Descrizione
`CurrentPlayerSessions`	Le sessioni giocatore con stato ATTIVO (il giocatore è connesso a una sessione di gioco attiva) o stato RISERVATO (al giocatore è stato assegnato uno slot in una sessione di gioco ma non è ancora stato connesso). e può essere utilizzato con la scalabilità automatica. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima
`PlayerSessionActivations`	Le sessioni giocatore che passano dallo stato RISERVATO a quello ATTIVO dall'ultimo rapporto. Questo succede quando un giocatore si connette correttamente a una sessione di gioco attiva. Unità: numero CloudWatch Statistiche pertinenti: somma, media, minima, massima

Parametri Amazon GameLift Servers per code

Il namespace `Amazon GameLift` include i seguenti parametri relativi all'attività in una coda di posizionamento delle sessioni di gioco. Le code vengono utilizzate con una soluzione Amazon GameLift Servers gestita. Il Amazon GameLift Servers servizio invia metriche a CloudWatch ogni minuto.

Parametro	Descrizione
`AverageWaitTime`	Il tempo medio di attesa prima che le richieste di posizionamento delle sessioni di gioco in coda con stato IN SOSPESO vengano soddisfatte. Unità: secondi CloudWatch Statistiche pertinenti: media, minima, massima, somma Dimensioni: ubicazione
`FirstChoiceNotViable`	Sessioni di gioco correttamente piazzate ma NON nel parco istanze di prima scelta, perché tale parco istanze non era considerato realizzabile (come un parco istanze Spot con un alto tasso di interruzione). Questa metrica si basa sul costo, non sulla latenza. La flotta di prima scelta è la prima flotta elencata in coda oppure, quando una richiesta di posiziona mento include dati sulla latenza dei giocatori, è la prima flotta scelta in base alla priorità. FleetIQ Se non ci sono parchi Spot realizzabili, potrebbe essere selezionato qualsiasi parco istanze in quella regione. Unità: numero Statistiche pertinenti: media, minima, massima, somma CloudWatch
`FirstChoiceOutOfCapacity`	Sessioni di gioco correttamente piazzate ma NON nel parco istanze di prima scelta, perché tale parco

Parametro	Descrizione
	istanze non aveva risorse disponibili. La flotta di prima scelta è la prima flotta elencata in coda oppure, quando una richiesta di posizionamento include dati sulla latenza dei giocatori, è la prima flotta scelta in base alla priorità definita. FleetIQ Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma
`LowestLatencyPlacement`	Le sessioni di gioco correttamente piazzate in una regione che offre la più bassa latenza possibile della coda per i giocatori. Questo parametro viene emesso solo quando i dati di latenza del giocatore sono inclusi nella richiesta di posizionamento. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma
`LowestPricePlacement`	Sessioni di gioco inserite con successo in una flotta al prezzo più basso possibile della coda per la regione selezionata. Il parco istanze può essere un parco istanze Spot o un'istanza on demand se la coda non presenta parchi istanze Spot. Questo parametro viene emesso solo quando i dati di latenza del giocatore sono inclusi nella richiesta di posiziona mento. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma

Parametro	Descrizione
`Placement <region name>`	Le sessioni di gioco correttamente piazzate nei parchi istanze posizionati nella regione specifica ta. Questo parametro scompone il parametro `PlacementsSucceeded` per regione. Unità: numero CloudWatchStatistiche pertinenti: somma
`PlacementsCanceled`	Le richieste di posizionamento delle sessioni di gioco che sono state annullate prima del timeout dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma
`PlacementsFailed`	Le richieste di posizionamento delle sessioni di gioco non andate a buon fine per qualsiasi motivo dall'ulti mo rapporto. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma
`PlacementsStarted`	Le nuove richieste di posizionamento delle sessioni di gioco aggiunte alla coda dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma

Parametro	Descrizione
`PlacementsSucceeded`	Le richieste di posizionamento delle sessioni di gioco che hanno avuto come esito una nuova sessione di gioco dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma
`PlacementsTimedOut`	Le richieste di posizionamento delle sessioni di gioco che hanno raggiunto il limite di timeout della coda senza venire soddisfatte dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma
`QueueDepth`	Il numero di richieste di posizionamento delle sessioni di gioco in coda con stato IN SOSPESO. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma Dimensioni: ubicazione

Parametri Amazon GameLift Servers per l'abbinamento

Lo spazio dei nomi `Amazon GameLift Servers` include i parametri dell'attività FlexMatch per le configurazioni di abbinamento e le regole di abbinamento. L'abbinamento FlexMatch viene utilizzato con una soluzione Amazon GameLift Servers gestita. Il Amazon GameLift Servers servizio invia metriche a CloudWatch ogni minuto.

Per ulteriori informazioni sulla sequenza delle attività di matchmaking, vedi Come funziona Amazon GameLift ServersFlexMatch.

Configurazioni di abbinamento

Parametro	Descrizione
`CurrentTickets`	Le richieste di abbinamento attualmente in fase di elaborazione o in attesa di essere elaborate. Unità: numero CloudWatch Statistiche pertinenti: media, minima, massima, somma
`MatchAcceptancesTimedOut`	Per le configurazioni di abbinamento che richiedon o accettazione, i potenziali abbinamenti scaduti durante l'accettazione dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: somma
`MatchesAccepted`	Per le configurazioni di abbinamento che richiedono accettazione, i potenziali abbinamenti che sono stati accettati dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: Sum
`MatchesCreated`	I potenziali abbinamenti che sono stati creati dall'ulti mo rapporto. Unità: numero CloudWatch Statistiche pertinenti: Sum
`MatchesPlaced`	Gli abbinamenti che sono stati posizionati correttam ente in una sessione di gioco dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: Sum

Parametro	Descrizione
`MatchesRejected`	Per le configurazioni di abbinamento che richiedono accettazione, i potenziali abbinamenti che sono stati rifiutati da almeno un giocatore dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: Sum
`PlayersStarted`	I giocatori in ticket di abbinamento che sono stati aggiunti dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: Sum
`TicketsFailed`	Le richieste di abbinamento che hanno prodotto un errore dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: Sum
`TicketsStarted`	Le nuove richieste di abbinamento che sono state create dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: Sum
`TicketsTimedOut`	Le richieste di abbinamento che hanno raggiunto il limite di timeout dall'ultimo rapporto. Unità: numero CloudWatch Statistiche pertinenti: Sum

Parametro	Descrizione
`TimeToMatch`	Per le richieste di abbinamento che vengono inserite in un abbinamento potenziale prima dell'ultimo rapporto, il tempo che intercorre tra la creazione del ticket e la creazione di abbinamenti potenziali. Unità: secondi CloudWatch Statistiche pertinenti: campioni di dati, media, minima, massima
`TimeToTicketCancel`	Per le richieste di abbinamento che sono state annullate prima dell'ultimo rapporto, il tempo che intercorre tra la creazione e l'annullamento del ticket. Unità: secondi CloudWatch Statistiche pertinenti: campioni di dati, media, minima, massima
`TimeToTicketSuccess`	Per le richieste di abbinamento che sono state completate prima dell'ultimo rapporto, il tempo che intercorre tra la creazione del ticket e il corretto posizionamento dell'abbinamento. Unità: secondi CloudWatch Statistiche pertinenti: campioni di dati, media, minima, massima

Regole di abbinamento

Parametro	Descrizione
`RuleEvaluationsPassed`	Le valutazioni delle regole durante il processo di abbinamento superato dall'ultimo rapporto. Questo parametro si limita alle prime 50 regole.

Parametro	Descrizione
	Unità: numero CloudWatchStatistiche pertinenti: somma
`RuleEvaluationsFailed`	Le valutazioni delle regole durante il processo di abbinamento non superato dall'ultimo rapporto. Questo parametro si limita alle prime 50 regole. Unità: numero CloudWatch Statistiche pertinenti: Sum

Parametri Amazon GameLift Servers per FleetIQ

Lo spazio dei nomi `Amazon GameLift Servers` include i parametri per il gruppo di server di gioco FleetIQ e l'attività del server di gioco come parte di una soluzione standalone FleetIQ per l'hosting di giochi. Il Amazon GameLift Servers servizio invia metriche a CloudWatch ogni minuto. Consulta anche Monitoraggio dei gruppi e delle istanze di Auto Scaling con Amazon CloudWatch nella Amazon EC2 Auto Scaling User Guide.

Parametro	Descrizione
`AvailableGameServers`	Server di gioco disponibili per eseguire un'esecuzione di un gioco non attualmente occupati dal gameplay. Questo numero include i server di gioco che sono stati registrati ma sono ancora in stato DISPONIBILE. Unità: numero CloudWatch Statistiche Amazon pertinenti: Sum Dimensioni: GameServerGroup
`UtilizedGameServers`	Server di gioco che sono attualmente occupati con gameplay. Questo numero include i server di gioco che sono nello stato UTILIZED (UTILIZZATO).

Parametro	Descrizione
	Unità: numero CloudWatch Statistiche Amazon pertinenti: Sum Dimensioni: GameServerGroup
`DrainingAvailableGameServers`	Server di gioco su istanze programmate per la terminazione che attualmente non supportano il gameplay. Questi server di gioco sono la priorità più bassa da rivendicare in risposta a una nuova richiesta di registrazione. Unità: numero CloudWatch Statistiche Amazon pertinenti: Sum Dimensioni: GameServerGroup
`DrainingUtilizedGameServers`	Server di gioco su istanze programmate per la terminazione che attualmente supportano la modalità di gioco. Unità: numero CloudWatch Statistiche Amazon pertinenti: Sum Dimensioni: GameServerGroup

Parametro	Descrizione
`PercentUtilizedGameServers`	Porzione di server di gioco che attualmente supportano le esecuzioni di gioco. Questa metrica indica la quantità di capacità del server di gioco attualmente in uso. È utile per guidare una politica di ridimensionamento automatico che può aggiunger e e rimuovere dinamicamente istanze in base alla domanda dei giocatori. Unità: percentuale CloudWatch Statistiche Amazon pertinenti: media, minima, massima Dimensioni: GameServerGroup
`GameServerInterruptions`	Server di gioco su istanze Spot che sono state interrotte a causa della disponibilità Spot limitata. Unità: numero CloudWatch Statistiche Amazon pertinenti: Sum Dimensioni: GameServerGroup, InstanceType
`InstanceInterruptions`	Istanze Spot che sono state interrotte a causa della disponibilità limitata. Unità: numero CloudWatch Statistiche Amazon pertinenti: Sum Dimensioni: GameServerGroup, InstanceType

Registrazione Amazon GameLift Servers Chiamate API con AWS CloudTrail

Amazon GameLift Servers è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in Amazon GameLift Servers. CloudTrail acquisisce tutte le chiamate API per Amazon GameLift Servers come eventi. Le chiamate acquisite includono chiamate provenienti da Amazon GameLift Servers console e chiamate in codice verso Amazon GameLift Servers operazioni API. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi eventi per Amazon GameLift Servers. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console in Cronologia eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta effettuata a Amazon GameLift Servers, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

Amazon GameLift Servers informazioni in CloudTrail

CloudTrail è abilitato sul tuo account al Account AWS momento della creazione dell'account. Quando si verifica un'attività in Amazon GameLift Servers, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti in Account AWS. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo Account AWS, inclusi gli eventi per Amazon GameLift Servers, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutti Amazon GameLift Servers le azioni vengono registrate CloudTrail e documentate nel Amazon GameLift Servers Riferimento API. Ad esempio`CreateGameSession`, le chiamate `CreatePlayerSession` e `UpdateGameSession` le azioni generano voci nei file di CloudTrail registro.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Comprensione Amazon GameLift Servers voci dei file di registro

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra le azioni `CreateFleet` e`DescribeFleetAttributes`.

```
{
    "Records": [
        {
            "eventVersion": "1.04",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/myUserName",
                "accountId": "111122223333",
                "accessKeyId": AKIAIOSFODNN7EXAMPLE",
                "userName": "myUserName"
            },
```

```
            "eventTime": "2015-12-29T23:40:15Z",
            "eventSource": "gamelift.amazonaws.com",
            "eventName": "CreateFleet",
            "awsRegion": "us-west-2",
            "sourceIPAddress": "192.0.2.0",
            "userAgent": "[]",
            "requestParameters": {
                "buildId": "build-92b6e8af-37a2-4c10-93bd-4698ea23de8d",
                "eC2InboundPermissions": [
                    {
                        "ipRange": "10.24.34.0/23",
                        "fromPort": 1935,
                        "protocol": "TCP",
                        "toPort": 1935
                    }
                ],
                "logPaths": [
                    "C:\\game\\serverErr.log",
                    "C:\\game\\serverOut.log"
                ],
                "eC2InstanceType": "c5.large",
                "serverLaunchPath": "C:\\game\\MyServer.exe",
                "description": "Test fleet",
                "serverLaunchParameters": "-paramX=baz",
                "name": "My_Test_Server_Fleet"
            },
            "responseElements": {
                "fleetAttributes": {
                    "fleetId": "fleet-0bb84136-4f69-4bb2-bfec-a9b9a7c3d52e",
                    "serverLaunchPath": "C:\\game\\MyServer.exe",
                    "status": "NEW",
                    "logPaths": [
                        "C:\\game\\serverErr.log",
                        "C:\\game\\serverOut.log"
                    ],
                    "description": "Test fleet",
                    "serverLaunchParameters": "-paramX=baz",
                    "creationTime": "Dec 29, 2015 11:40:14 PM",
                    "name": "My_Test_Server_Fleet",
                    "buildId": "build-92b6e8af-37a2-4c10-93bd-4698ea23de8d"
                }
            },
            "requestID": "824a2a4b-ae85-11e5-a8d6-61d5cafb25f2",
            "eventID": "c8fbea01-fbf9-4c4e-a0fe-ad7dc205ce11",
```

```
            "eventType": "AwsApiCall",
            "recipientAccountId": "111122223333"
        },
        {
            "eventVersion": "1.04",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/myUserName",
                "accountId": "111122223333",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "userName": "myUserName"
            },
            "eventTime": "2015-12-29T23:40:15Z",
            "eventSource": "gamelift.amazonaws.com",
            "eventName": "DescribeFleetAttributes",
            "awsRegion": "us-west-2",
            "sourceIPAddress": "192.0.2.0",
            "userAgent": "[]",
            "requestParameters": {
                "fleetIds": [
                    "fleet-0bb84136-4f69-4bb2-bfec-a9b9a7c3d52e"
                ]
            },
            "responseElements": null,
            "requestID": "82e7f0ec-ae85-11e5-a8d6-61d5cafb25f2",
            "eventID": "11daabcb-0094-49f2-8b3d-3a63c8bad86f",
            "eventType": "AwsApiCall",
            "recipientAccountId": "111122223333"
        },
    ]
}
```

Registrazione dei messaggi del server Amazon GameLift Servers

Puoi acquisire messaggi server personalizzati dai tuoi Amazon GameLift Servers server in file di registro. Il modo in cui si configura la registrazione dipende dal fatto che si utilizzino server personalizzati o Amazon GameLift Servers Realtime (vedere le sottosezioni appropriate in questo capitolo).

Argomenti

- Messaggi del server di registrazione () Amazon GameLift ServersRealtime

Messaggi del server di registrazione (server personalizzati)

È possibile acquisire messaggi server personalizzati dai server Amazon GameLift Servers personalizzati in file di registro. Per ulteriori informazioni sulla registrazione di Amazon GameLift ServersRealtime, consultaMessaggi del server di registrazione () Amazon GameLift ServersRealtime.

> **Important**
>
> C'è un limite alla dimensione di un file di registro per sessione di gioco (vedi Amazon GameLift Serversendpoint e quote in). Riferimenti generali di AWS Al termine di una sessione di gioco, Amazon GameLift Servers carica i log del server su Amazon Simple Storage Service (Amazon S3). Amazon GameLift Serversnon caricherà log che superano il limite. I log possono crescere molto rapidamente e superare il limite di dimensione. È necessario monitorare i log e limitare l'output dei log ai soli messaggi necessari.

Configurazione della registrazione per server personalizzati

Con i server Amazon GameLift Servers personalizzati, è possibile scrivere il proprio codice per eseguire la registrazione, che viene configurato come parte della configurazione del processo del server. Amazon GameLift Serversutilizza la tua configurazione di registrazione per identificare i file che deve caricare su Amazon S3 al termine di ogni sessione di gioco.

Le seguenti istruzioni mostrano come configurare la registrazione utilizzando esempi di codice semplificati:

C++

Per configurare la registrazione (C++)

1. Crea un vettore di stringhe che siano percorsi di directory verso i file di registro del server di gioco.

```
std::string serverLog("serverOut.log");        // Example server log file
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);
```

2. Fornisci il tuo vettore come immagine LogParametersdel tuo oggetto. ProcessParameters

```
Aws::GameLift::Server::ProcessParameters processReadyParameter =
 Aws::GameLift::Server::ProcessParameters(
    std::bind(&Server::onStartGameSession, this, std::placeholders::_1),
    std::bind(&Server::onProcessTerminate, this),
    std::bind(&Server::OnHealthCheck, this),
    std::bind(&Server::OnUpdateGameSession, this),
    listenPort,
    Aws::GameLift::Server::LogParameters(logPaths));
```

3. Fornisci l'ProcessParametersoggetto quando chiami ProcessReady().

```
Aws::GameLift::GenericOutcome outcome =
   Aws::GameLift::Server::ProcessReady(processReadyParameter);
```

Per un esempio più completo, vediProcessReady().

C#

Per configurare la registrazione (C#)

1. Crea un elenco di stringhe che sono percorsi di directory verso i file di registro del server di gioco.

```
List<string> logPaths = new List<string>();
logPaths.Add("C:\\game\\serverOut.txt");     // Example of a log file that the
 game server writes
```

2. Fornisci la tua lista come ProcessParametersoggetto. LogParameters

```
var processReadyParameter = new ProcessParameters(
    this.OnGameSession,
    this.OnProcessTerminate,
    this.OnHealthCheck,
    this.OnGameSessionUpdate,
    port,
    new LogParameters(logPaths));
```

3. Fornisci l'ProcessParametersoggetto quando chiami ProcessReady().

```
var processReadyOutcome =
```

```
    GameLiftServerAPI.ProcessReady(processReadyParameter);
```

Per un esempio più completo, vediProcessReady().

Scrittura nei log

I file di registro esistono dopo l'avvio del processo del server. È possibile scrivere nei log utilizzando qualsiasi metodo di scrittura sui file. Per acquisire tutti gli output standard e gli output di errore del server, rimappa i flussi di output nei file di registro, come negli esempi seguenti:

C++

```
std::freopen("serverOut.log", "w+", stdout);
std::freopen("serverErr.log", "w+", stderr);
```

C#

```
Console.SetOut(new StreamWriter("serverOut.txt"));
Console.SetError(new StreamWriter("serverErr.txt"));
```

Accesso ai log del server

Al termine di una sessione di gioco, archivia Amazon GameLift Servers automaticamente i log in un bucket Amazon S3 e li conserva per 14 giorni. Per ottenere la posizione dei log di una sessione di gioco, puoi utilizzare l'operazione API. GetGameSessionLogUrl Per scaricare i log, utilizza l'URL restituito dall'operazione.

Messaggi del server di registrazione () Amazon GameLift ServersRealtime

Puoi acquisire messaggi personalizzati dal server dai tuoi file Amazon GameLift Servers Realtime di registro. Per ulteriori informazioni sulla registrazione per server personalizzati, consultaMessaggi del server di registrazione (server personalizzati).

Esistono diversi tipi di messaggi che puoi inviare ai tuoi file di registro (vedi). Registrazione dei messaggi nello script del server Oltre ai messaggi personalizzati, il sistema di Amazon GameLift Servers Realtime output utilizza gli stessi tipi di messaggi e scrive negli stessi file di registro. Puoi regolare il livello di registrazione del tuo parco macchine per ridurre la quantità di messaggi di registrazione generati dai server (vediRegolazione del livello di registrazione).

> **Important**
>
> C'è un limite alla dimensione di un file di registro per sessione di gioco (vedi Amazon GameLift Serversendpoint e quote nella). Riferimenti generali di AWS Al termine di una sessione di gioco, Amazon GameLift Servers carica i log del server su Amazon Simple Storage Service (Amazon S3). Amazon GameLift Serversnon caricherà log che superano il limite. I log possono crescere molto rapidamente e superare il limite di dimensione. È necessario monitorare i log e limitare l'output dei log ai soli messaggi necessari.

Registrazione dei messaggi nello script del server

Puoi inviare messaggi personalizzati nello script per il tuo Amazon GameLift ServersRealtime. Utilizza i seguenti passaggi per inviare messaggi del server a un file di registro:

1. Create una variabile per contenere il riferimento all'oggetto logger.

   ```
   var logger;
   ```

2. Nella `init()` funzione, recupera il logger dall'oggetto di sessione e assegnalo alla variabile logger.

   ```
   function init(rtSession) {
       session = rtSession;
       logger = session.getLogger();
   }
   ```

3. Chiama la funzione appropriata sul logger per emettere un messaggio.

   Messaggi di debug

   ```
   logger.debug("This is my debug message...");
   ```

   Messaggi informativi

   ```
   logger.info("This is my info message...");
   ```

   Messaggi di avviso

```
logger.warn("This is my warn message...");
```

Messaggi di errore

```
logger.error("This is my error message...");
```

Messaggi di errore irreversibili

```
logger.fatal("This is my fatal error message...");
```

Il cliente riceve messaggi di errore irreversibili

```
logger.cxfatal("This is my customer experience fatal error message...");
```

Per un esempio delle istruzioni di registrazione in uno script, vedere. Amazon GameLift ServersRealtimeesempio di script

L'output nei file di registro indica il tipo di messaggio (`DEBUG`,,`INFO`,`WARN`,`ERROR`,`CXFATAL`)`FATAL`, come illustrato nelle righe seguenti di un registro di esempio:

```
09 Sep 2021 11:46:32,970 [INFO] (gamelift.js) 215: Calling GameLiftServerAPI.InitSDK...
09 Sep 2021 11:46:32,993 [INFO] (gamelift.js) 220: GameLiftServerAPI.InitSDK succeeded
09 Sep 2021 11:46:32,993 [INFO] (gamelift.js) 223: Waiting for Realtime server to
 start...
09 Sep 2021 11:46:33,15 [WARN] (index.js) 204: Connection is INSECURE. Messages will be
 sent/received as plaintext.
```

Accesso ai log del server

Al termine di una sessione di gioco, archivia Amazon GameLift Servers automaticamente i log in Amazon S3 e li conserva per 14 giorni. Puoi utilizzare la chiamata GetGameSessionLogUrl API per ottenere la posizione dei log di una sessione di gioco. Utilizza l'URL restituito dalla chiamata API per scaricare i log.

Regolazione del livello di registrazione

I registri possono crescere molto rapidamente e superare il limite di dimensione. È necessario monitorare i log e limitare l'output dei log ai soli messaggi necessari. In Amazon GameLift Servers Realtime effetti, puoi regolare il livello di registrazione fornendo un parametro nella configurazione di runtime del tuo parco macchine nel modulo`loggingLevel:`*`LOGGING_LEVEL`*, dove *`LOGGING_LEVEL`* è presente uno dei seguenti valori:

1. `debug`
2. `info`(impostazione predefinita)
3. `warn`
4. `error`
5. `fatal`
6. `cxfatal`

Questo elenco è ordinato dal meno grave (`debug`) al più grave (`cxfatal`). Se ne `loggingLevel` imposti uno, il server registrerà solo i messaggi con quel livello di gravità o un livello di gravità superiore. Ad esempio, l'impostazione `loggingLevel:error` farà in modo che tutti i server del parco utenti `error` scrivano e `cxfatal` inviino solo messaggi nel registro. `fatal`

Puoi impostare il livello di registrazione per la tua flotta al momento della creazione o dopo l'esecuzione. La modifica del livello di registrazione della flotta dopo l'avvio avrà effetto solo sui registri delle sessioni di gioco create dopo l'aggiornamento. I registri di tutte le sessioni di gioco esistenti non verranno modificati. Se non imposti un livello di registrazione quando crei la tua flotta, i tuoi server imposteranno il livello di registrazione come predefinito. `info` Consulta le seguenti sezioni per istruzioni su come impostare il livello di registrazione.

Impostazione del livello di registrazione durante la creazione di un Amazon GameLift ServersRealtime parco veicoli (console)

Segui le istruzioni riportate Crea una EC2 flotta Amazon GameLift Servers gestita per creare la tua flotta, con la seguente aggiunta:

- Nella sottofase di allocazione del processo Server della fase Gestione del processo, fornisci la coppia chiave-valore del livello di registrazione (ad esempio**`loggingLevel:error`**) come valore per i parametri di Launch. Utilizzate un carattere non alfanumerico (eccetto la virgola) per separare

il livello di registrazione da eventuali parametri aggiuntivi (ad esempio,). `loggingLevel:error +map Winter444`

Impostazione del livello di registrazione durante la creazione di una flotta () Amazon GameLift ServersRealtimeAWS CLI

Segui le istruzioni riportate Crea una EC2 flotta Amazon GameLift Servers gestita per creare la tua flotta, con la seguente aggiunta:

- Nell'argomento relativo al `--runtime-configuration` parametro for`create-fleet`, fornite la coppia chiave-valore del livello di registrazione (ad esempio`loggingLevel:error`) come valore per. `Parameters` Utilizzate un carattere non alfanumerico (eccetto la virgola) per separare il livello di registrazione da eventuali parametri aggiuntivi. Fai riferimento al file di esempio seguente:

```
--runtime-configuration "GameSessionActivationTimeoutSeconds=60,
                    MaxConcurrentGameSessionActivations=2,
                    ServerProcesses=[{LaunchPath=/local/game/myRealtimeLaunchScript.js,
                                      Parameters=loggingLevel:error +map Winter444,
                                      ConcurrentExecutions=10}]"
```

Impostazione del livello di registrazione per un parco macchine funzionante (console) Amazon GameLift ServersRealtime

Segui le istruzioni riportate Aggiornamento di una configurazione del Amazon GameLift Servers parco veicoli per aggiornare il tuo parco veicoli utilizzando la Amazon GameLift Servers Console, con la seguente aggiunta:

- Nella pagina Modifica flotta, in Allocazione dei processi del server, fornisci la coppia chiave-valore del livello di registrazione (ad esempio**`loggingLevel:error`**) come valore per i parametri di Launch. Utilizza un carattere non alfanumerico (eccetto la virgola) per separare il livello di registrazione da eventuali parametri aggiuntivi (ad esempio,). `loggingLevel:error +map Winter444`

Impostazione del livello di registrazione per una flotta in funzione () Amazon GameLift ServersRealtimeAWS CLI

Segui le istruzioni riportate Aggiornamento di una configurazione del Amazon GameLift Servers parco veicoli per aggiornare la tua flotta utilizzando AWS CLI, con la seguente aggiunta:

- Nell'argomento relativo al `--runtime-configuration` parametro for[update-runtime-configuration](), fornite la coppia chiave-valore del livello di registrazione (ad esempio`loggingLevel:error`) come valore per. `Parameters` Utilizzate un carattere non alfanumerico (eccetto la virgola) per separare il livello di registrazione da eventuali parametri aggiuntivi. Fai riferimento al file di esempio seguente:

```
--runtime-configuration "GameSessionActivationTimeoutSeconds=60,
                   MaxConcurrentGameSessionActivations=2,
                   ServerProcesses=[{LaunchPath=/local/game/myRealtimeLaunchScript.js,
                                  Parameters=loggingLevel:error +map Winter444,
                                  ConcurrentExecutions=10}]"
```

Sicurezza in Amazon GameLift Servers

Se stai usando Amazon GameLift Servers FleetIQ come funzionalità autonoma di Amazon EC2, consulta la sezione Security in Amazon EC2 nella Amazon EC2 User Guide.

La sicurezza del cloud AWS è la massima priorità. In quanto cliente AWS , puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra te AWS e te. Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. Per saperne di più sui programmi di conformità che si applicano a Amazon GameLift Servers, vedi AWS i servizi rientranti nell'ambito del programma di conformità, .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è responsabile anche di altri fattori, tra cui la sensibilità dei dati, i requisiti aziendali AWS e le normative applicabili.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo Amazon GameLift Servers. Nei seguenti argomenti viene illustrato come configurare Amazon GameLift Servers per raggiungere i tuoi obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere i Amazon GameLift Servers risorse.

Argomenti

Protezione dei dati in Amazon GameLift Servers

Se utilizzi Amazon GameLift Servers FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Security in Amazon EC2 nella Amazon EC2 User Guide.

Il modello di responsabilità AWS condivisa modello si applica alla protezione dei dati inAmazon GameLift Servers. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori Amazon GameLift Servers o Servizi AWS utilizzi la console, l'API o. AWS CLI AWS

SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

I dati specifici di Amazon GameLift Servers vengono gestiti come segue:

- Le build dei server di gioco e gli script su cui carichi Amazon GameLift Servers vengono archiviati in Amazon S3. Non vi è alcun accesso diretto del cliente a questi dati una volta caricati. Un utente autorizzato può ottenere l'accesso temporaneo al caricamento dei file, ma non può visualizzare o aggiornare i file direttamente in Amazon S3. Per eliminare script e build, utilizza la console Amazon GameLift Servers o l'API del servizio.
- I dati di registro delle sessioni di gioco vengono archiviati in Amazon S3 per un periodo di tempo limitato dopo il completamento della sessione di gioco. Gli utenti autorizzati possono accedere ai dati di log scaricandoli tramite un collegamento nella console Amazon GameLift Servers o tramite chiamate all'API del servizio.
- I dati relativi a parametri ed eventi vengono archiviati in Amazon GameLift Servers ed è possibile accedervi tramite la console Amazon GameLift Servers o le chiamate all'API del servizio. I dati possono essere recuperati su parchi istanze, istanze, posizionamenti di sessioni di gioco, ticket di matchmaking, sessioni di gioco e sessioni di giocatori. È possibile accedere ai dati anche tramite Amazon CloudWatch ed CloudWatch Events.
- I dati forniti dal cliente vengono memorizzati in Amazon GameLift Servers. Gli utenti autorizzati possono accedervi tramite chiamate all'API del servizio. I dati potenzialmente sensibili possono includere i dati del giocatore, i dati della sessione del giocatore e della sessione di gioco (incluse le informazioni sulla connessione), i dati di matchmaker e così via.

> **Note**
>
> Se inserisci un giocatore personalizzato IDs nelle tue richieste, è previsto che questi valori siano resi anonimi UUIDs e non contengano informazioni identificative del giocatore.

Per ulteriori informazioni sulla protezione dei dati, consulta il post del blog relativo al modello di responsabilità condivisa AWS e GDPR in AWS Security Blog.

Crittografia dei dati inattivi

La crittografia dei dati inattivi specifici di Amazon GameLift Servers viene gestita come segue:

- Le build e gli script dei server di gioco sono archiviati in bucket Amazon S3 con crittografia lato server.
- I dati forniti dal cliente vengono archiviati in Amazon GameLift Servers in formato crittografato.

Crittografia in transito

Le connessioni a Amazon GameLift Servers APIs vengono effettuate tramite una connessione sicura (SSL) e autenticate utilizzando la versione 4 di AWS Signature (quando ci si connette tramite AWS CLI o AWS SDK, la firma viene gestita automaticamente). L'autenticazione viene gestita utilizzando le policy di accesso definite da IAM per le credenziali di sicurezza utilizzate per effettuare la connessione.

La comunicazione diretta tra i client di gioco e i server di gioco è la seguente:

- Per i server di gioco personalizzati ospitati in risorse Amazon GameLift Servers, la comunicazione non coinvolge il servizio Amazon GameLift Servers. La crittografia di questa comunicazione è una responsabilità del cliente. È possibile utilizzare parchi istanze abilitati per TLS per consentire ai client di gioco di autenticare il server di gioco durante la connessione e per crittografare tutte le comunicazioni tra il client di gioco e il server di gioco.
- Infatti, Amazon GameLift Servers Realtime con la generazione di certificati TLS abilitata, il traffico tra il client di gioco e i Realtime server che utilizza il client SDK for viene crittografato durante il trasferimento. Realtime Il traffico TCP viene crittografato con TLS 1.2 e il traffico UDP viene crittografato con DTLS 1.2.

Riservatezza del traffico Internet

È possibile accedere in remoto alle istanze Amazon GameLift Servers in modo sicuro. Per le istanze che utilizzano Linux, SSH fornisce un canale di comunicazione sicuro per l'accesso remoto. Per le istanze che eseguono Windows, utilizza un client RDP (Remote Desktop Protocol). Con Amazon GameLift ServersFleetIQ, l'accesso remoto alle istanze tramite AWS Systems Manager Session Manager e Run Command è crittografato utilizzando TLS 1.2 e le richieste per creare una connessione vengono firmate utilizzando SigV4. Per informazioni sulla connessione a un'istanza Amazon GameLift Servers gestita, consulta Connessione remota a Amazon GameLift Servers flotte di istanze.

Gestione delle identità e degli accessi per Amazon GameLift Servers

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) all'uso Amazon GameLift Servers risorse. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che svolgi Amazon GameLift Servers.

Utente del servizio: se si utilizza il Amazon GameLift Servers per svolgere il proprio lavoro, l'amministratore fornisce all'utente le credenziali e le autorizzazioni necessarie. Man mano che ne usi di più Amazon GameLift Servers per eseguire il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non è possibile accedere a una funzionalità in Amazon GameLift Servers, consulta Risoluzione dei problemi Amazon GameLift Servers identità e accesso.

Amministratore del servizio: se sei responsabile di Amazon GameLift Servers le risorse della tua azienda, probabilmente hai pieno accesso a Amazon GameLift Servers. Spetta a te determinare quale Amazon GameLift Servers funzionalità e risorse a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a

IAM. Per saperne di più su come la tua azienda può utilizzare IAM con Amazon GameLift Servers, consulta In che modo Amazon GameLift Servers funziona con IAM.

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso a Amazon GameLift Servers. Per visualizzare un esempio Amazon GameLift Servers politiche basate sull'identità che puoi utilizzare in IAM, vedi. Esempi di policy basate sull'identità per Amazon GameLift Servers

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo consigliato per la firma delle richieste, consulta Signature Version 4 AWS per le richieste API nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA)AWS in IAM nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti

alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi passare da un ruolo utente a un ruolo IAM (console). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Create a role for a third-party identity provider (federation) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso inoltrato (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. È preferibile archiviare le chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella IAM User Guide.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Elenchi di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo del servizio (SCPs): SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più di

proprietà dell' Account AWS azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS Per ulteriori informazioni su Organizations and SCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.

- Politiche di controllo delle risorse (RCPs): RCPs sono politiche JSON che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le politiche IAM allegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations e RCPs, incluso un elenco di Servizi AWS tale supporto RCPs, vedere Resource control policies (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta la logica di valutazione delle policy nella IAM User Guide.

In che modo Amazon GameLift Servers funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a Amazon GameLift Servers, scopri con quali funzionalità IAM è possibile utilizzare Amazon GameLift Servers.

Funzionalità IAM che puoi utilizzare con Amazon GameLift Servers

Funzionalità IAM	Amazon GameLift Servers supporto
Policy basate su identità	Sì

Funzionalità IAM	Amazon GameLift Servers supporto
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
ACLs	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una visione di alto livello di come Amazon GameLift Servers e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Politiche basate sull'identità per Amazon GameLift Servers

Supporta le policy basate su identità: sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica

all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di politiche basate sull'identità per Amazon GameLift Servers

Per visualizzare esempi di Amazon GameLift Servers politiche basate sull'identità, vedere. Esempi di policy basate sull'identità per Amazon GameLift Servers

politiche basate sulle risorse all'interno Amazon GameLift Servers

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Azioni politiche per Amazon GameLift Servers

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione

AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per un elenco di Amazon GameLift Servers azioni, vedi Azioni definite da Amazon GameLift Serversnel riferimento di autorizzazione del servizio.

Azioni politiche in Amazon GameLift Servers usa il seguente prefisso prima dell'azione:

```
gamelift
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "gamelift:action1",
      "gamelift:action2"
         ]
```

È possibile specificare più operazioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Describe`, includi la seguente azione:

```
"Action": "gamelift:Describe*"
```

Per visualizzare esempi di Amazon GameLift Servers politiche basate sull'identità, vedere. Esempi di policy basate sull'identità per Amazon GameLift Servers

Risorse politiche per Amazon GameLift Servers

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È

possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le operazioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per un elenco di Amazon GameLift Servers tipi di risorse e relativi ARNs, vedere Risorse definite da Amazon GameLift Servers nel riferimento di autorizzazione del servizio. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, vedere Azioni definite da Amazon GameLift Servers.

Medio Amazon GameLift Servers le risorse hanno valori ARN, che consentono di gestire l'accesso alle risorse utilizzando le policy IAM. Il Amazon GameLift Servers fleet resource ha un ARN con la seguente sintassi:

```
arn:${Partition}:gamelift:${Region}:${Account}:fleet/${FleetId}
```

Per ulteriori informazioni sul formato di ARNs, consulta Amazon Resource Names (ARNs) nel Riferimenti generali di AWS.

Ad esempio, per specificare il parco istanze `fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa` nell'istruzione, utilizza il seguente ARN:

```
"Resource": "arn:aws:gamelift:us-west-2:123456789012:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa"
```

Per specificare tutte le flotte che appartengono a un account specifico, usa un carattere jolly (*):

```
"Resource": "arn:aws:gamelift:us-west-2:123456789012:fleet/*"
```

Per visualizzare esempi di Amazon GameLift Servers politiche basate sull'identità, vedere. Esempi di policy basate sull'identità per Amazon GameLift Servers

Chiavi relative alle condizioni delle politiche per Amazon GameLift Servers

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per un elenco di Amazon GameLift Servers chiavi di condizione, vedi Chiavi di condizione per Amazon GameLift Servernel riferimento di autorizzazione del servizio. Per sapere con quali azioni e risorse è possibile utilizzare una chiave di condizione, consulta Azioni definite da Amazon GameLift Servers.

Per visualizzare esempi di Amazon GameLift Servers politiche basate sull'identità, vedere. Esempi di policy basate sull'identità per Amazon GameLift Servers

ACLs in Amazon GameLift Servers

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con Amazon GameLift Servers

Supporta ABAC (tag nelle policy): sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Per un esempio di politica basata sull'identità che limita l'accesso a una risorsa in base ai tag presenti su quella risorsa, vedi. Vista Amazon GameLift Servers flotte basate su tag

Utilizzo di credenziali temporanee con Amazon GameLift Servers

Supporta le credenziali temporanee: sì

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente

e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Passaggio da un ruolo utente a un ruolo IAM (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni principali multiservizio per Amazon GameLift Servers

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per Amazon GameLift Servers

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompersi Amazon GameLift Servers funzionalità. Modifica i ruoli di servizio solo quando Amazon GameLift Servers fornisce indicazioni per farlo.

Consenti il tuo Amazon GameLift Servers-server di gioco ospitati per accedere ad altre AWS risorse, come una AWS Lambda funzione o un database Amazon DynamoDB. Perché i server di gioco sono ospitati su flotte che Amazon GameLift Servers gestisce, è necessario un ruolo di servizio che dia

Amazon GameLift Servers accesso limitato alle altre AWS risorse. Per ulteriori informazioni, consulta Comunica con altre AWS risorse delle tue flotte.

ruoli collegati ai servizi per Amazon GameLift Servers

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per i dettagli sulla creazione o la gestione di ruoli collegati ai servizi, consulta i AWS servizi che funzionano con IAM nella IAM User Guide. Trova un servizio nella tabella che include un `Yes` nella colonna Ruoli collegati ai servizi. Scegli Sì per visualizzare la documentazione del ruolo collegato al servizio per quel servizio.

Esempi di policy basate sull'identità per Amazon GameLift Servers

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare Amazon GameLift Servers risorse. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da Amazon GameLift Servers, incluso il formato di ARNs per ogni tipo di risorsa, vedi Azioni, risorse e chiavi di condizione per Amazon GameLift Servers nel riferimento di autorizzazione del servizio.

Argomenti

- Best practice per le policy
- Utilizzo di Amazon GameLift Servers console
- Consentire agli utenti di visualizzare le loro autorizzazioni
- Consentire l'accesso ai giocatori per le sessioni di gioco
- Consentire l'accesso a uno Amazon GameLift Servers coda

- Vista Amazon GameLift Servers flotte basate su tag
- Accedi a un file di build del gioco in Amazon S3

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare Amazon GameLift Servers risorse presenti nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA

quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo di Amazon GameLift Servers console

Per accedere al Amazon GameLift Servers console, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentire di elencare e visualizzare i dettagli relativi a Amazon GameLift Servers risorse presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Per garantire che tali entità possano ancora utilizzare il Amazon GameLift Servers console, aggiungi le autorizzazioni a utenti e gruppi con la sintassi riportata negli esempi seguenti e in. Esempi di autorizzazioni amministrative Per ulteriori informazioni, consulta Imposta le autorizzazioni utente per Amazon GameLift Servers.

Utenti che collaborano con Amazon GameLift Servers le operazioni AWS CLI tramite le AWS nostre API non richiedono autorizzazioni minime per la console. Puoi invece limitare l'accesso solo alle operazioni che l'utente deve eseguire. Ad esempio, un utente giocatore, che agisce per conto dei client di gioco, richiede l'accesso per richiedere sessioni di gioco, inserire giocatori nei giochi e altre attività.

Per informazioni sulle autorizzazioni necessarie per utilizzare tutto Amazon GameLift Servers funzionalità della console, vedere la sintassi delle autorizzazioni per gli amministratori in. Esempi di autorizzazioni amministrative

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

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

```
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Consentire l'accesso ai giocatori per le sessioni di gioco

Per inserire i giocatori nelle sessioni di gioco, i client di gioco e i servizi di backend necessitano di autorizzazioni. Per esempi di policy per questi scenari, consulta. Esempi di autorizzazioni utente per i giocatori

Consentire l'accesso a uno Amazon GameLift Servers coda

L'esempio seguente fornisce a un utente l'accesso a uno specifico Amazon GameLift Servers code.

Questa politica concede all'utente le autorizzazioni per aggiungere, aggiornare ed eliminare le destinazioni della coda con le seguenti azioni:`gamelift:UpdateGameSessionQueue`, e. `gamelift:DeleteGameSessionQueue` `gamelift:DescribeGameSessionQueues` Come

illustrato, questo criterio utilizza l'`Resource`elemento per limitare l'accesso a una singola coda:. `gamesessionqueue/examplequeue123`

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ViewSpecificQueueInfo",
         "Effect":"Allow",
         "Action":[
            "gamelift:DescribeGameSessionQueues"
         ],
         "Resource":"arn:aws:gamelift:::gamesessionqueue/examplequeue123"
      },
      {
         "Sid":"ManageSpecificQueue",
         "Effect":"Allow",
         "Action":[
            "gamelift:UpdateGameSessionQueue",
            "gamelift:DeleteGameSessionQueue"
         ],
         "Resource":"arn:aws:gamelift:::gamesessionqueue/examplequeue123"
      }
   ]
}
```

Vista Amazon GameLift Servers flotte basate su tag

Puoi utilizzare le condizioni della tua politica basata sull'identità per controllare l'accesso a Amazon GameLift Servers risorse basate su tag. Questo esempio mostra come è possibile creare una politica che consenta di visualizzare una flotta se il `Owner` tag corrisponde al nome utente dell'utente. Questa politica concede anche le autorizzazioni necessarie per completare questa operazione nella console.

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

```
        {
            "Sid": "ViewFleetIfOwner",
            "Effect": "Allow",
            "Action": "gamelift:DescribeFleetAttributes",
            "Resource": "arn:aws:gamelift:*:*:fleet/*",
            "Condition": {
                "StringEquals": {"gamelift:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Accedi a un file di build del gioco in Amazon S3

Dopo aver integrato il server di gioco con Amazon GameLift Servers, carica i file di build su Amazon S3. In Amazon GameLift Servers per accedere ai file di build, utilizza la seguente politica.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::bucket-name/object-name"
        }
    ]
}
```

Per ulteriori informazioni sul caricamento Amazon GameLift Servers file di gioco, vediImplementa una build di server personalizzata per Amazon GameLift Servers hosting.

Risoluzione dei problemi Amazon GameLift Servers identità e accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con Amazon GameLift Servers e AWS Identity and Access Management (IAM).

Argomenti

Non sono autorizzato a eseguire un'azione in Amazon GameLift Servers

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, contatta l'amministratore del tuo AWS account per ricevere assistenza. L'amministratore è colui che ti ha fornito le credenziali di accesso.

Il seguente errore di esempio si verifica quando l'utente `mateojackson` IAM tenta di utilizzare la console per visualizzare i dettagli su una coda ma non dispone delle `gamelift:DescribeGameSessionQueues` autorizzazioni:

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 gamelift:DescribeGameSessionQueues on resource: examplequeue123
```

In questo caso, Mateo chiede al suo amministratore di aggiornare le sue politiche per consentirgli l'accesso in lettura alla `examplequeue123` risorsa utilizzando l'azione. `gamelift:DescribeGameSessionQueues`

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di assegnare un ruolo a Amazon GameLift Servers.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in Amazon GameLift Servers. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di assistenza, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS al mio Amazon GameLift Servers risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon GameLift Servers supporta queste funzionalità, vediIn che modo Amazon GameLift Servers funziona con IAM.
- Per scoprire come fornire l'accesso alle risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide. Account AWS
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

AWS politiche gestite per Amazon GameLift Servers

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consulta Policy gestite da AWSnella Guida per l'utente di IAM.

AWS politica gestita: GameLiftContainerFleetPolicy

Puoi collegarti `GameLiftContainerFleetPolicy` ai tuoi ruoli IAM.

La policy concede le autorizzazioni per le azioni di calcolo in un Amazon GameLift Servers flotta di container. Una flotta di container è un insieme di risorse di hosting che Amazon GameLift Servers gestisce per te. Amazon GameLift Servers necessita delle autorizzazioni per connettersi al Amazon GameLift Servers servizio e altri AWS servizi per tuo conto.

Quando si crea una flotta di container con Amazon GameLift Servers, fornisci un ruolo di servizio IAM con la policy GameLiftContainerFleetPolicy gestita allegata. Per istruzioni sulla creazione del ruolo di servizio, consultaConfigurare un ruolo di servizio IAM per Amazon GameLift Servers.

Per ulteriori informazioni, consulta GameLiftContainerFleetPolicy.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `cloudwatch`— Permette Amazon GameLift Servers per scrivere i log delle sessioni di gioco in un flusso di log di Amazon CloudWatch Events nel tuo AWS account.
- `cloudwatch` — Permette Amazon GameLift Servers per creare un gruppo di CloudWatch log per organizzare i dati della sessione di gioco nel flusso di log.
- `s3`— Permette Amazon GameLift Servers per scrivere i log delle sessioni di gioco in un bucket Amazon Simple Storage Service nel tuo AWS account.
- `s3`— Permette Amazon GameLift Servers per recuperare la posizione Regione AWS in cui risiede un bucket Amazon S3 specificato, utilizzando l'azione API. `s3:GetBucketLocation`

- `gamelift` — Permette Amazon GameLift Servers di recuperare un token di autenticazione che consenta a un server di gioco ospitato di comunicare con Amazon GameLift Servers servizio tramite il tuo AWS account.

Amazon GameLift Servers aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon GameLift Servers da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS sul Amazon GameLift Servers Pagina delle note di rilascio.

Modifica	Descrizione	Data
GameLiftContainerFleetPolicy— Modifica	Amazon GameLift Servers ha aggiunto nuove autorizzazioni per recuperare i dati Regione AWS di un bucket Amazon S3.	5 febbraio 2024
GameLiftContainerFleetPolicy: nuova policy	Amazon GameLift Servers ha aggiunto nuove autorizzazioni per consentire l'esecuzione dei contenitori dei server di gioco Amazon GameLift Servers flotte gestite.	12 novembre 2024
Amazon GameLift Servers ha iniziato a tracciare le modifiche	Amazon GameLift Servers ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	12 novembre 2024

Registrazione e monitoraggio con Amazon GameLift Servers

Il monitoraggio è una parte importante del mantenimento dell'affidabilità, della disponibilità e delle prestazioni di Amazon GameLift Servers e le tue AWS soluzioni. È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica.

AWS e Amazon GameLift Servers forniscono diversi strumenti per monitorare le risorse di hosting dei giochi e rispondere a potenziali incidenti.

CloudWatch Allarmi Amazon

Utilizzando Amazon CloudWatch alarms, controlli una singola metrica per un periodo di tempo specificato. Se la metrica supera una determinata soglia, viene inviata una notifica a un argomento o a una politica di Auto AWS Scaling di Amazon SNS. CloudWatch gli allarmi vengono attivati quando il loro stato cambia e vengono mantenuti per un determinato numero di periodi, non perché si trovano in uno stato particolare. Per ulteriori informazioni, consulta Monitora Amazon GameLift Servers con Amazon CloudWatch.

AWS CloudTrail Registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon GameLift Servers. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata a Amazon GameLift Servers, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi. Per ulteriori informazioni, consulta Registrazione Amazon GameLift Servers Chiamate API con AWS CloudTrail.

Convalida della conformità per Amazon GameLift Servers

Amazon GameLift Servers non rientra nell'ambito di alcun programma di AWS conformità.

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Governance e conformità per la sicurezza: queste guide all'implementazione di soluzioni illustrano considerazioni relative all'architettura e i passaggi per implementare le funzionalità di sicurezza e conformità.

- Riferimenti sui servizi conformi ai requisiti HIPAA: elenca i servizi HIPAA idonei. Non tutti Servizi AWS sono idonei alla normativa HIPAA.
- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- Valutazione delle risorse con regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, come lo standard PCI DSS, soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente l' AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Resilienza in Amazon GameLift Servers

Se stai usando Amazon GameLift Servers FleetIQ come funzionalità autonoma di Amazon EC2, consulta la sezione Security in Amazon EC2 nella Amazon EC2 User Guide.

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta infrastruttura globale.AWS

Oltre all'infrastruttura AWS globale, Amazon GameLift Servers offre le seguenti funzionalità per supportare le esigenze di resilienza dei dati:

- Code multiregionali: Amazon GameLift Servers le code delle sessioni di gioco vengono utilizzate per inserire nuove sessioni di gioco con le risorse di hosting disponibili. Le code che si estendono su più regioni sono in grado di reindirizzare i posizionamenti delle sessioni di gioco in caso di interruzione regionale. Per ulteriori informazioni e le best practice sulla creazione di code di sessioni di gioco, consulta Personalizza una coda di sessioni di gioco.
- Scalabilità automatica della capacità: mantieni l'integrità e la disponibilità delle risorse di hosting utilizzando Amazon GameLift Servers strumenti di ridimensionamento. Questi strumenti offrono una vasta gamma di opzioni che consentono di regolare la capacità del parco istanze in base alle esigenze del gioco e dei giocatori. Per ulteriori informazioni sul dimensionamento, consultare Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers.
- Distribuzione tra le istanze: Amazon GameLift Servers distribuisce il traffico in entrata su più istanze, a seconda delle dimensioni del parco veicoli. Come best practice, i giochi in produzione devono avere più istanze per mantenere la disponibilità nel caso in cui un'istanza risulti non integra o non risponde.
- Storage Amazon S3: build e script di server di gioco caricati su Amazon GameLift Servers sono archiviati in Amazon S3 utilizzando la classe di storage Standard, che utilizza più repliche di data center per aumentare la resilienza. I log delle sessioni di gioco vengono inoltre archiviati in Amazon S3 utilizzando la classe di storage Standard.

Sicurezza dell'infrastruttura in Amazon GameLift Servers

Se stai usando Amazon GameLift Servers FleetIQ come funzionalità autonoma di Amazon EC2, consulta la sezione Security in Amazon EC2 nella Amazon EC2 User Guide.

Come servizio gestito, Amazon GameLift Servers è protetto dalle procedure di sicurezza della rete AWS globale descritte nel white paper Amazon Web Services: panoramica dei processi di sicurezza.

Utilizzi chiamate API AWS pubblicate per accedere Amazon GameLift Servers attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. È consigliabile TLS 1.3 o versioni successive. I client devono, inoltre, supportare le suite di crittografia con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral

Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Il Amazon GameLift Servers il servizio colloca tutte le flotte nei cloud privati virtuali di Amazon (VPCs) in modo che ogni flotta esista in un'area logicamente isolata nel AWS cloud. È possibile utilizzare… Amazon GameLift Servers policy per controllare l'accesso da endpoint VPC specifici o specifici. VPCs In effetti, questo isola l'accesso alla rete a un determinato Amazon GameLift Servers risorsa proveniente solo dal VPC specifico all'interno della AWS rete. Quando crei un parco istanze specifici un intervallo di numeri di porta e indirizzi IP. Questi intervalli limitano il modo in cui il traffico in entrata può accedere ai server di gioco ospitati nel VPC di un parco istanze. Utilizza le best practice di sicurezza standard per scegliere le impostazioni di accesso del parco istanze.

Analisi della configurazione e delle vulnerabilità in Amazon GameLift Servers

Se stai usando Amazon GameLift Servers FleetIQ come funzionalità autonoma di Amazon EC2, consulta la sezione Security in Amazon EC2 nella Amazon EC2 User Guide.

Configurazione e controllo IT sono una responsabilità condivisa tra AWS e te, il nostro cliente. Per ulteriori informazioni, consulta il modello di responsabilità AWS condivisa. AWS gestisce le attività di sicurezza di base come l'applicazione di patch al sistema operativo guest (OS) e al database, la configurazione del firewall e il disaster recovery. Queste procedure sono state riviste e certificate dalle terze parti appropriate. Per ulteriori dettagli, consulta la seguente risorsa: Amazon Web Services: panoramica dei processi di sicurezza (white paper).

Le seguenti best practice di sicurezza riguardano anche la configurazione e l'analisi delle vulnerabilità in Amazon GameLift Servers:

- I clienti sono responsabili della gestione del software distribuito su Amazon GameLift Servers istanze per l'hosting di giochi. Nello specifico:
  - Il software applicativo del server di gioco fornito dal cliente deve essere gestito, inclusi gli aggiornamenti e le patch di sicurezza. Per aggiornare il software del server di gioco, carica una nuova build su Amazon GameLift Servers, crea una nuova flotta e reindirizza il traffico verso la nuova flotta.

- L'Amazon Machine Image (AMI) di base che include il sistema operativo, viene aggiornata solo quando viene creato un nuovo parco istanze. Per applicare patch, aggiornare e proteggere il sistema operativo e altre applicazioni che fanno parte dell'AMI, ricicla regolarmente i parchi istanze, indipendentemente dagli aggiornamenti del server di gioco.
- I clienti dovrebbero prendere in considerazione la possibilità di aggiornare regolarmente i propri giochi con le ultime versioni dell'SDK, tra cui AWS SDK, Amazon GameLift Servers Server SDK e Amazon GameLift Servers Client SDK per server in tempo reale.

Best practice relative alla sicurezza di Amazon GameLift Servers

Se utilizzi Amazon GameLift Servers FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Security in Amazon EC2 nella Amazon EC2 User Guide.

Amazon GameLift Servers fornisce una serie di caratteristiche di sicurezza che occorre valutare durante lo sviluppo e l'implementazione delle policy di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per l'ambiente, gestiscile come considerazioni utili anziché prescrizioni.

Non aprire porte di accesso a Internet

Consigliamo vivamente di non aprire porte a Internet perché ciò comporta un rischio per la sicurezza. Ad esempio, se si utilizza UpdateFleetPortSettingsper aprire una porta desktop remota come questa:

```
{
  "FleetId": "<fleet identifier>",
  "InboundPermissionAuthorizations": [
      {
        "FromPort": 3389,
        "IpRange": "0.0.0.0/0",
        "Protocol": "RDP",
        "ToPort": 3389
      }
  ]
}
```

allora stai permettendo a chiunque su Internet di accedere all'istanza.

Invece, apri la porta con un indirizzo IP specifico o un intervallo di indirizzi. Ad esempio, in questo modo:

```
{
  "FleetId": "<fleet identifier>",
  "InboundPermissionAuthorizations": [
      {
        "FromPort": 3389,
        "IpRange": "54.186.139.221/32",
        "Protocol": "TCP",
        "ToPort": 3389
      }
  ]
}
```

Ulteriori informazioni

Per ulteriori informazioni su come utilizzare in modo Amazon GameLift Servers più sicuro, consulta il pilastro AWS Well-Architected Tool Sicurezza. .

Amazon GameLift Serversguide di riferimento

Questa sezione contiene la documentazione di riferimento per l'utilizzo Amazon GameLift Servers.

Argomenti

API di servizio per Amazon GameLift Servers

Usa questo elenco basato sulle attività per trovare le operazioni API durante la creazione di soluzioni di hosting di Amazon GameLift Servers giochi e altre funzionalità. L' AWS SDK include queste operazioni nel namespace. `aws.gamelift` Scarica l' AWS SDK o visualizza la documentazione di riferimento dell'Amazon GameLift ServersAPI. Puoi anche utilizzare l'API con l'interfaccia a riga di AWS comando (AWS CLI), come documentato nel riferimento ai AWS CLI comandi.

L'API include due serie di operazioni per l'hosting gestito di giochi:

- Gestisci le risorse Amazon GameLift Servers di hosting
- Inizia sessioni di gioco e unisciti ai giocatori

L'API Amazon GameLift Servers di servizio contiene anche operazioni da utilizzare con altri Amazon GameLift Servers strumenti e soluzioni. Per un elenco di FleetIQ APIs, consulta Operazioni FleetIQ API. Per un elenco di FlexMatch APIs matchmaking, vedi Operazioni FlexMatch API.

Gestisci le risorse Amazon GameLift Servers di hosting

Richiama queste operazioni per configurare le risorse di hosting per i tuoi server di gioco, scalare la capacità per soddisfare la domanda dei giocatori, ottenere metriche di prestazioni e utilizzo e altro

ancora. Utilizza queste operazioni API per ospitare server di gioco conAmazon GameLift Servers, incluso Amazon GameLift ServersRealtime. Puoi anche lavorare Amazon GameLift Serverssulla console per la maggior parte delle attività di gestione delle risorse oppure puoi effettuare chiamate con lo strumento AWS Command Line Interface (AWS CLI).

Prepara i server di gioco per la distribuzione

Carica e configura il codice del server di gioco del gioco in preparazione della distribuzione e del lancio sulle risorse di hosting.

Gestisci build di server di gioco personalizzate

- upload-build: carica i file di build da un percorso locale e crea una nuova Amazon GameLift Servers risorsa di build. Questa operazione, disponibile come AWS CLI comando, è il modo più comune per caricare build di server di gioco.
- CreateBuild— Crea una nuova build utilizzando file archiviati in un bucket Amazon S3.
- ListBuilds— Ottieni un elenco di tutte le build caricate in una regione. Amazon GameLift Servers
- DescribeBuild— Recupera le informazioni associate a una build.
- UpdateBuild— Modifica i metadati della build, inclusi il nome e la versione della build.
- DeleteBuild— Rimuovi una build daAmazon GameLift Servers.

Gestisci gli script Amazon GameLift Servers Realtime di configurazione

- CreateScript— Carica JavaScript file e crea una nuova risorsa di Amazon GameLift Servers script.
- ListScripts— Ottieni un elenco di tutti gli Realtime script caricati in una Amazon GameLift Servers regione.
- DescribeScript— Recupera le informazioni associate a uno Realtime script.
- UpdateScript— Modifica i metadati dello script e carica il contenuto dello script rivisto.
- DeleteScript— Rimuovere uno Realtime script da. Amazon GameLift Servers

Configura le risorse informatiche per l'hosting

Configura le risorse di hosting e distribuiscile con lo script di build o Realtime configurazione del server di gioco.

Crea e gestisci flotte

- CreateFleet— Configura e distribuisci una nuova Amazon GameLift Servers flotta di risorse informatiche per far funzionare i tuoi server di gioco. Una volta distribuiti, i server di gioco vengono avviati automaticamente così come configurati e pronti per ospitare sessioni di gioco.
- ListFleets— Ottieni un elenco di tutte le flotte in una Amazon GameLift Servers regione.
- DeleteFleet— Rimuovi una flotta che non utilizza più server di gioco o ospita giocatori.
- Visualizza/aggiorna le posizioni della flotta.
  - CreateFleetLocations— Aggiungi sedi remote a una flotta esistente che supporta più sedi
  - DescribeFleetLocationAttributes— Ottieni un elenco di tutte le località remote di una flotta e visualizza lo stato attuale di ciascuna sede.
  - DeleteFleetLocations— Rimuovi le postazioni remote da una flotta che supporta più sedi.
- Visualizzare/aggiornare le configurazioni del parco istanze.
  - DescribeFleetAttributes/UpdateFleetAttributes— Visualizza o modifica i metadati e le impostazioni di una flotta per la protezione delle sessioni di gioco e i limiti di creazione di risorse.
  - DescribeFleetPortSettings/UpdateFleetPortSettings— Visualizza o modifica le autorizzazioni in entrata (indirizzi IP e intervalli di impostazione delle porte) consentite per un parco veicoli.
  - DescribeRuntimeConfiguration/UpdateRuntimeConfiguration— Visualizza o modifica quali processi server (e quanti) eseguire su ciascuna istanza di un parco istanze.

Gestisci la capacità del parco veicoli

- Descrivi EC2 InstanceLimits: recupera il numero massimo di istanze consentite per l' AWS account corrente e il livello di utilizzo corrente.
- DescribeFleetCapacity— Recupera le impostazioni di capacità correnti per la regione di origine di una flotta.
- DescribeFleetLocationCapacity— Recupera le impostazioni di capacità correnti per ogni sede di una flotta con più sedi.
- UpdateFleetCapacity— Regola manualmente le impostazioni di capacità per una flotta.
- Configurazione:
  - PutScalingPolicy— Attiva l'auto-scaling basato sull'obiettivo o crea una politica di auto-scaling personalizzata o aggiorna una politica esistente.
  - DescribeScalingPolicies— Recuperare una politica di auto-scaling esistente.
  - DeleteScalingPolicy— Eliminare una politica di auto-scaling e impedire che influisca sulla capacità della flotta.

- StartFleetActions— Riavvia le politiche di auto-scaling di una flotta.
- StopFleetActions— Sospendere le politiche di auto-scaling di una flotta.

Monitorare l'attività del parco istanze.

- DescribeFleetUtilization— Recupera le statistiche sul numero di processi del server, sessioni di gioco e giocatori attualmente attivi su una flotta.
- DescribeFleetLocationUtilization— Recupera le statistiche di utilizzo per ogni località in una flotta con più sedi.
- DescribeFleetEvents— Visualizza gli eventi registrati per una flotta durante un periodo di tempo specificato.
- DescribeGameSessions— Recupera i metadati della sessione di gioco, tra cui la durata di una partita e il numero attuale di giocatori.

Imposta le code per il posizionamento delle sessioni di gioco

Configura le code in più regioni e in più parchi istanze per posizionare le sessioni di gioco con le migliori risorse di hosting disponibili per costo, latenza e resilienza.

- CreateGameSessionQueue— Crea una coda da utilizzare durante l'elaborazione delle richieste di posizionamento nelle sessioni di gioco.
- DescribeGameSessionQueues— Recupera le code delle sessioni di gioco definite in una regione. Amazon GameLift Servers
- UpdateGameSessionQueue— Modifica la configurazione di una coda di sessione di gioco.
- DeleteGameSessionQueue— Rimuovi una coda di sessione di gioco dalla regione.

Gestione di alias

Utilizza gli alias per rappresentare i parchi istanze o per creare una destinazione alternativa del terminale. Gli alias sono utili per la transizione dell'attività di gioco da un parco istanze all'altro, ad esempio durante gli aggiornamenti della build del server di gioco.

- CreateAlias— Definisci un nuovo alias e, facoltativamente, assegnalo a una flotta.
- ListAliases— Ottieni tutti gli alias della flotta definiti in una regione. Amazon GameLift Servers
- DescribeAlias— Recupera informazioni su un alias esistente.

- UpdateAlias— Modifica le impostazioni di un alias, ad esempio reindirizzandolo da una flotta all'altra.
- DeleteAlias— Rimuovere un alias dalla regione.
- ResolveAlias— Ottieni l'ID della flotta a cui punta un alias specificato.

Connect a istanze di hosting gestite

Visualizza le informazioni sulle singole istanze di un parco istanze o richiedi l'accesso remoto a una specifica istanza del parco istanze per la risoluzione dei problemi.

- DescribeInstances— Ottieni informazioni su ogni istanza del parco istanze, tra cui ID dell'istanza, indirizzo IP, posizione e stato.
- GetInstanceAccess— Richiedere le credenziali di accesso necessarie per connettersi in remoto a un'istanza specifica in un parco istanze.

Configurazione del peering VPC

Crea e gestisci connessioni peering VPC tra le tue risorse di Amazon GameLift Servers hosting e altre risorse. AWS

- CreateVpcPeeringAuthorization— Autorizza una connessione peering a uno dei tuoi. VPCs
- DescribeVpcPeeringAuthorizations— Recupera autorizzazioni di connessione peering valide.
- DeleteVpcPeeringAuthorization— Eliminare l'autorizzazione di una connessione peering.
- CreateVpcPeeringConnection— Stabilisci una connessione peering tra il VPC di Amazon GameLift Servers una flotta e uno dei tuoi. VPCs
- DescribeVpcPeeringConnections— Recupera informazioni sulle connessioni peering VPC attive o in sospeso con una flotta. Amazon GameLift Servers
- DeleteVpcPeeringConnection— Eliminare una connessione peering VPC con una flotta. Amazon GameLift Servers

Inizia sessioni di gioco e unisciti ai giocatori

Richiama queste operazioni da un servizio di backend per avviare nuove sessioni di gioco, ottenere informazioni sulle sessioni di gioco esistenti e unire i giocatori alle sessioni di gioco. Queste operazioni sono destinate all'uso con server di gioco personalizzati ospitati suAmazon

GameLift Servers. Se utilizzi Amazon GameLift ServersRealtime, gestisci le sessioni di gioco utilizzandoAmazon GameLift ServersRealtimeriferimento all'API client (C#).

- Avviare nuove sessioni di gioco per uno o più giocatori.
  - StartGameSessionPlacement— Chiedi Amazon GameLift Servers di trovare le migliori risorse di hosting disponibili e di iniziare una nuova sessione di gioco. Questo è il metodo preferito per creare nuove sessioni di gioco. Si basa sulle code delle sessioni di gioco per tenere traccia della disponibilità di hosting in più regioni e utilizza FleetIQ algoritmi per dare priorità ai posizionamenti in base alla latenza dei giocatori, ai costi di hosting, alla posizione, ecc.
  - DescribeGameSessionPlacement— Ottieni dettagli e stato di una richiesta di collocamento.
  - StopGameSessionPlacement— Annullare una richiesta di collocamento.
  - CreateGameSession— Inizia una nuova sessione di gioco vuota in una posizione specifica della flotta. Questa operazione ti dà un maggiore controllo su dove iniziare la sessione di gioco, anziché utilizzarla FleetIQ per valutare le opzioni di posizionamento. È necessario aggiungere giocatori alla nuova sessione di gioco in un passaggio separato.
- Attira i giocatori nelle sessioni di gioco esistenti. Trova sessioni di gioco in corso con gli slot disponibili e riservale ai nuovi giocatori.
  - CreatePlayerSession— Riserva uno slot libero per consentire a un giocatore di partecipare a una sessione di gioco.
  - CreatePlayerSessions— Riserva slot aperti per consentire a più giocatori di partecipare a una sessione di gioco.
- Utilizzare dati sulle sessioni di gioco e dei giocatori. Gestisci le informazioni sulle sessioni di gioco e sulle sessioni dei giocatori.
  - SearchGameSessions— Richiedi un elenco di sessioni di gioco attive in base a una serie di criteri di ricerca.
  - DescribeGameSessions— Recupera i metadati per sessioni di gioco specifiche, tra cui la durata di attività e il numero attuale di giocatori.
  - DescribeGameSessionDetails— Recupera i metadati, inclusa l'impostazione di protezione della sessione di gioco, per una o più sessioni di gioco.
  - DescribePlayerSessions— Ottieni dettagli sull'attività dei giocatori, tra cui stato, tempo di gioco e dati del giocatore.
  - UpdateGameSession— Modifica le impostazioni della sessione di gioco, ad esempio il numero massimo di giocatori e la politica di iscrizione.
  - GetGameSessionLogUrl— Ottieni la posizione dei log salvati per una sessione di gioco.

Server SDK 5.x per Amazon GameLift Servers

Questa sezione fornisce la documentazione di riferimento per il server SDK 5.x for. Amazon GameLift Servers L'SDK del server fornisce le funzionalità di base utilizzate dal server di gioco per interagire con il servizio. Amazon GameLift Servers Ad esempio, il server di gioco riceve dal servizio richieste di avvio e interruzione delle sessioni di gioco e fornisce aggiornamenti regolari sullo stato delle sessioni di gioco al servizio. Integra i tuoi server di gioco con l'SDK del server prima di distribuirli per l'hosting.

Usa questo riferimento all'SDK del server per integrare i tuoi server di gioco multiplayer personalizzati con cui ospitare. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

L'ultima versione principale del server SDK for Amazon GameLift Servers è la 5.x. Le seguenti funzionalità di hosting richiedono l'uso della versione 5.x:

- Amazon GameLift ServersOvunque
- Amazon GameLift Serversplugin per Unreal Engine e Unity

Argomenti

Aggiornamenti nel server SDK 5 per Amazon GameLift Servers

I server di gioco ospitati utilizzano l'SDK del server Amazon GameLift Servers per comunicare con il Amazon GameLift Servers servizio e avviare e gestire le sessioni di gioco per i giocatori. L'ultima versione, Amazon GameLift Servers server SDK 5, offre una serie di miglioramenti e supporto per nuove Amazon GameLift Servers funzionalità. Se la versione del tuo server di gioco utilizza attualmente Amazon GameLift Servers server SDK 4 o versioni precedenti, segui le indicazioni riportate in questo argomento per aggiornare i tuoi giochi.

Amazon GameLift Serversla versione 5.0.0 e successive dell'SDK del server include questi aggiornamenti:

- Linguaggi espansi: le librerie sono disponibili nelle seguenti lingue: C++, C#, Go. Puoi creare le librerie C++ da utilizzare con Unreal Engine.
- Supporto per i plug-in del motore di gioco: i plug-in Amazon GameLift Servers autonomi per Unreal Engine e Unity richiedono Amazon GameLift Servers le librerie SDK 5 del server. Questi plugin offrono flussi di lavoro guidati per l'integrazione, il test e la distribuzione dei giochi per l'hosting. Amazon GameLift Servers Vedi e documentazione. Amazon GameLift Serversplugin per Unity (server SDK 5.x) Amazon GameLift Serversplugin per Unreal Engine
- Amazon GameLift ServersSupporto ovunque: con le flotte Anywhere puoi configurare le tue risorse di hosting per utilizzare le Amazon GameLift Servers funzionalità (incluso il matchmaking). Aggiungi l'Amazon GameLift Serversagente per automatizzare la gestione del ciclo di vita delle sessioni di gioco. Usa le flotte Anywhere per l'hosting di produzione con hardware locale o configura ambienti di test per un rapido sviluppo iterativo di giochi. See Hosting ovunque and the Agent. Amazon GameLift Servers
- Strumenti di test aggiornati: la funzione Amazon GameLift Servers Anywhere ti consente di configurare ambienti di test locali o basati sul cloud per i tuoi giochi. Configura i test con o senza l'Amazon GameLift Serversagente. Questi strumenti sostituiscono Amazon GameLift Servers Local. Consultare Configura i test locali con Amazon GameLift Servers Anywhere.
- Soluzione .NET consolidata per C#: il server C# SDK 5.1+ supporta .NET Framework 4.6.2 (aggiornato dalla versione 4.6.1) e.NET 6.0 in un'unica soluzione. .NET Standard 2.1 è disponibile con le librerie create da Unity.
- Nuova `Compute` risorsa: questa nuova risorsa combina diversi tipi di risorse di hosting. Include risorse di hosting basate sul cloud (flotte gestite EC2 o container) e risorse di hosting controllate dal cliente (flotte Anywhere). Include i seguenti aggiornamenti:
  - Le nuove chiamate API per la `Compute` risorsa includono: ListCompute(), DescribeCompute() e GetComputeAccess(). Queste azioni restituiscono informazioni sulle risorse di hosting per qualsiasi tipo di Amazon GameLift Servers flotta. In generale, per le flotte con server di gioco che utilizzano il server SDK 5.x, utilizza le azioni specifiche del computer per sostituire le azioni specifiche dell'istanza. Inoltre, queste azioni possono essere utilizzate nelle flotte Anywhere senza l'Amazon GameLift Serversagente: RegisterCompute(), () e (). DeregisterCompute GetComputeAuthToken
  - Nuova metrica `ActiveCompute` con CloudWatch dimensioni `FleetIdLocation`, e. `ComputeType` Questa metrica sostituisce la metrica precedente. `ActiveInstances`

- Amazon EC2 Systems Manager (SSM) per l'accesso remoto: per una maggiore sicurezza, utilizza SSM anziché SSH per la connessione a istanze in flotte gestite. Amazon GameLift Servers Consultare Connessione remota a Amazon GameLift Servers flotte di istanze.

Migrazione al server SDK 5.x per Amazon GameLift Servers

Per aggiornare un progetto di gioco per utilizzare la versione 5.x del server SDK, apporta le seguenti modifiche:

1. Scarica l'SDK del server più recente per il Amazon GameLift Servers pacchetto per il tuo ambiente di sviluppo [Scarica il sito]. Segui le istruzioni di installazione contenute nel `Readme` file per il pacchetto e la versione scaricati. Consulta queste istruzioni per utilizzare il server SDKs con il tuo progetto di gioco.

   - Per ambienti di sviluppo che utilizzano C++, C# o Go
   - Per progetti Unreal Engine (SDK del server C++ solo per le librerie Unreal)
   - Per i progetti Unity (SDK del server C# solo per le librerie Unity)
   - Da utilizzare con il Amazon GameLift Servers plugin per Unreal Engine
   - Da usare con il Amazon GameLift Servers plugin per Unity

2. Aggiorna il codice del server come segue:

   - Cambia la funzione di callback del codice server `onCreateGameSession()` in`onStartGameSession()`.
   - Aggiorna gli `InitSDK()` input in modo appropriato:
     - Se hai intenzione di distribuire il server di gioco, costruisci una EC2 flotta Amazon GameLift Servers gestita o una flotta Anywhere con l'Amazon GameLift Serversagente:

       Chiama `InitSDK()` senza parametri (C++) (C#) (Unreal). Questa chiamata configura l'ambiente di calcolo e una WebSocket connessione al servizio. Amazon GameLift Servers
     - Se hai intenzione di utilizzare il server di gioco, crea una flotta Anywhere senza l'Amazon GameLift Serversagente:

       Chiama `InitSDK()` con i parametri del server (C++) (C#) (Unreal). Un processo del server di gioco utilizza questi parametri per stabilire una connessione con il servizio. Amazon GameLift Servers

3. Se la build del server di gioco o altre applicazioni ospitate comunicano con altre AWS risorse durante l'esecuzione, dovrai modificare il modo in cui l'applicazione accede a tali risorse. Sostituisci l'uso di `AssumeRoleCredentials` con il nuovo server SDK action `GetFleetRoleCredentials()` (per i server di gioco) o usa credenziali condivise (per altre applicazioni). Per ulteriori informazioni su come implementare questa modifica, consulta. Comunica con altre AWS risorse delle tue flotte
4. Se il progetto ha chiamato l'azione SDK del server `GetInstanceCertificate()` per recuperare un certificato TLS, modifica il codice per utilizzare invece il nuovo `GetComputeCertificate()` (C++) (C#) (Unreal).
5. Quando carichi la build del gioco su Amazon GameLift Servers (ad esempio con upload-build o CreateBuild()), imposta il `ServerSdkVersion` parametro sulla versione 5.x che stai utilizzando (questo parametro attualmente è predefinito su 4.0.2). Questo parametro deve corrispondere alle librerie SDK del server effettive nella build del server di gioco. Se specifichi la versione sbagliata per una build del server di gioco caricata, tutte le flotte create con quella build falliranno. Consultare Implementa una build di server personalizzata per Amazon GameLift Servers hosting.

   L'esempio seguente illustra come specificare la versione SDK del server:

   ```
   aws gamelift upload-build \
       --operating-system AMAZON_LINUX_2023 \
       --server-sdk-version "5.0.0" \
       --build-root "~/mygame" \
       --name "My Game Nightly Build" \
       --build-version "build 255" \
       --region us-west-2
   ```

6. Se utilizzi script per connetterti in remoto a flotte gestite, aggiorna gli script per utilizzare il nuovo processo, come descritto in. Connessione remota a Amazon GameLift Servers flotte di istanze

Server C++ SDK 5.x per -- Azioni Amazon GameLift Servers

Usa il riferimento all'SDK 5.x del server per integrare il tuo gioco multiplayer con cui ospitarlo. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consulta. Aggiungi Amazon GameLift Servers al tuo server di gioco

> **Note**
>
> Questo argomento descrive l'API Amazon GameLift Servers C++ che è possibile utilizzare quando si crea con la libreria standard C++ ()`std`. In particolare, questa documentazione si applica al codice compilato con l'opzione. `-DGAMELIFT_USE_STD=1`

Server C++ SDK 5.x per Amazon GameLift Servers -- Tipi di dati

Usa il riferimento all'SDK 5.x del server Amazon GameLift Servers C++ per integrare il tuo gioco multiplayer con cui ospitare. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consulta. Aggiungi Amazon GameLift Servers al tuo server di gioco

> **Note**
>
> Questo argomento descrive l'API Amazon GameLift Servers C++ che è possibile utilizzare quando si crea con la libreria standard C++ ()`std`. In particolare, questa documentazione si applica al codice compilato con l'opzione. `-DGAMELIFT_USE_STD=1`

Server C++ SDK 5.x per -- Azioni Amazon GameLift Servers

Tipi di dati

- LogParameters
- ProcessParameters
- UpdateGameSession
- GameSession
- ServerParameters
- StartMatchBackfillRequest
- Player
- DescribePlayerSessionsRequest
- StopMatchBackfillRequest
- AttributeValue
- GetFleetRoleCredentialsRequest

- AwsLongOutcome
- AwsStringOutcome
- DescribePlayerSessionsOutcome
- DescribePlayerSessionsResult
- GenericOutcome
- GenericOutcomeCallable
- PlayerSession
- StartMatchBackfillOutcome
- StartMatchBackfillResult
- GetComputeCertificateOutcome
- GetComputeCertificateResult
- GetFleetRoleCredentialsOutcome
- GetFleetRoleCredentialsResult
- Inizialo SDKOutcome
- GameLiftError
- enumerazioni;

LogParameters

Un oggetto che identifica i file generati durante una sessione di gioco che desideri caricare e archiviare Amazon GameLift Servers al termine della sessione di gioco. Il server di gioco `LogParameters` lo fornisce Amazon GameLift Servers come parte di un `ProcessParameters` oggetto in una ProcessReady() chiamata.

Proprietà	Descrizione
LogPaths	L'elenco dei percorsi di directory dei file di registro del server di gioco che desideri Amazon GameLift Servers archiviare per accessi futuri. Il processo del server genera questi file durante ogni sessione di gioco. Definisci i percorsi e i nomi dei file nel tuo server di gioco e li memorizzi nella directory di build del gioco principale.

	I percorsi di registro devono essere assoluti. Ad esempio, se la build del gioco memorizza i log delle sessioni di gioco in un percorso simile`MyGame\sessionLogs\` , il percorso si troverebbe `c:\game\MyGame\sessionLogs` su un'istanza di Windows. Tipo: `std:vector<std::string>` Required: No

ProcessParameters

Questo tipo di dati contiene l'insieme di parametri inviati a Amazon GameLift Servers in aProcessReady().

Proprietà	Descrizione
LogParameters	Un oggetto con percorsi di directory ai file generati durante una sessione di gioco. Amazon GameLift Serverscopia e archivia i file per accessi futuri. Tipo: `Aws::GameLift::Server:: LogParameters` Required: No
OnHealthCheck	La funzione di callback che Amazon GameLift Servers richiama per richiedere un rapporto sullo stato di salute del processo del server. Amazon GameLift Serverschiama questa funzione ogni 60 secondi e attende una risposta per 60 secondi. Il processo del server ritorna `TRUE` se integro, `FALSE` se non integro. Se non viene restituita alcuna risposta, Amazon GameLift Servers registra il processo del server come non integro.

	Tipo: `std::function<bool()> onHealthCheck` Required: No
OnProcessTerminate	La funzione di callback che Amazon GameLift Servers richiama per forzare l'arresto del processo server. Dopo aver chiamato questa funzione, Amazon GameLift Servers attende 5 minuti che il processo server si spenga e risponda con una ProcessEnding() chiamata prima di chiudere il processo server. Tipo: `std::function<void()> onProcessTerminate` Campo obbligatorio: sì
OnStartGameSession	La funzione di callback che Amazon GameLift Servers richiama l'attivazione di una nuova sessione di gioco. Amazon GameLift Serversch iama questa funzione in risposta a una richiesta del client. CreateGameSession La funzione di callback passa un GameSession oggetto. Tipo: `const std::function<void (Aws::GameLift::Model::Game Session)> onStartGameSession` Campo obbligatorio: sì

OnUpdateGameSession	La funzione di callback che Amazon GameLift Servers richiama per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift Serverschiama questa funzione quando è stata elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un GameSession oggetto, uno status update (`updateReason`) e l'ID del ticket match backfill. Tipo: `std::function<void(Aws::GameLift::Server::Model::UpdateGameSession)> onUpdateGameSession` Required: No
Porta	Il numero di porta su cui il processo server ascolta le connessioni di nuovi giocatori . Il valore deve rientrare nella gamma di porte configurate per qualsiasi parco istanze che distribuisce questa build del server di gioco. Questo numero di porta è incluso nella sessione di gioco e negli oggetti della sessione del giocatore utilizzati dalle sessioni di gioco per la connessione a un processo server. Tipo: `Integer` Campo obbligatorio: sì

UpdateGameSession

Questo tipo di dati viene aggiornato a un oggetto della sessione di gioco, che include il motivo per cui la sessione di gioco è stata aggiornata e il relativo ID del ticket di backfill se il backfill viene utilizzato per riempire le sessioni dei giocatori nella sessione di gioco.

Proprietà	Descrizione
GameSession	Oggetto GameSession. L'`GameSessi on` oggetto contiene proprietà che descrivono una sessione di gioco. Tipo: `Aws::GameLift::Server::Game Session` Campo obbligatorio: sì
UpdateReason	Il motivo per cui la sessione di gioco viene aggiornata. Tipo: `Aws::GameLift::Server::Upda teReason` Campo obbligatorio: sì
BackfillTicketId	L'ID del ticket di backfill che tenta di aggiornare la sessione di gioco. Tipo: `std::string` Required: No

GameSession

Questo tipo di dati fornisce i dettagli di una sessione di gioco.

Proprietà	Descrizione
GameSessionId	Un identificatore univoco per la sessione di gioco. L'ARN di una sessione di gioco ha il seguente formato:. `arn:aws:gamelift:< region>::gamesession/<fleet ID>/<custom ID string or idempoten cy token>`

Proprietà	Descrizione
	Tipo: `std::string` Required: No
Nome	Un'etichetta descrittiva della sessione di gioco. Tipo: `std::string` Required: No
FleetId	Un identificatore univoco per la flotta su cui è in esecuzione la sessione di gioco. Tipo: `std::string` Required: No
MaximumPlayerSessionCount	Il numero massimo di connessioni dei giocatori alla sessione di gioco. Tipo: `int` Required: No
Porta	Il numero di porta per la sessione di gioco. Per connettersi a un server di Amazon GameLift Servers gioco, un'app necessita sia dell'indi rizzo IP che del numero di porta. Tipo: `in` Required: No

Proprietà	Descrizione
IpAddress	L'indirizzo IP della sessione di gioco. Per connettersi a un server di Amazon GameLift Servers gioco, un'app necessita sia dell'indi rizzo IP che del numero di porta. Tipo: `std::string` Required: No
GameSessionData	Un set di proprietà personalizzate della sessione di gioco, formattate come valore di stringa singola. Tipo: `std::string` Required: No
MatchmakerData	Informazioni sul processo di matchmaking utilizzato per creare la sessione di gioco, in sintassi JSON, formattate come stringa. Oltre alla configurazione di matchmaking utilizzata, contiene dati su tutti i giocatori assegnati alla partita, compresi gli attributi dei giocatori e le assegnazioni delle squadre. Tipo: `std::string` Required: No
GameProperties	Un insieme di proprietà personalizzate per una sessione di gioco, formattate come coppie chiave:valore. Queste proprietà vengono passate con una richiesta di avvio di una nuova sessione di gioco. Tipo: `std :: vector < GameProperty >` Required: No

Proprietà	Descrizione
DnsName	L'identificatore DNS assegnato all'istanza che esegue la sessione di gioco. I valori hanno il seguente formato: • Flotte abilitate per TLS:. `<unique identifier>.<region identifier>.amazongamelift.com` • Non-TLS-enabled flotte:. `ec2-<unique identifier>.compute.amazonaws.com` Quando ti connetti a una sessione di gioco in esecuzione su una flotta abilitata per TLS, devi usare il nome DNS, non l'indirizzo IP. Tipo: `std::string` Required: No

ServerParameters

Informazioni utilizzate da un processo del server di gioco per stabilire una connessione con il servizio. Amazon GameLift Servers Includi questi parametri durante la chiamata InitSDK() solo se la build del server di gioco verrà distribuita su una flotta Anywhere o su una flotta di container senza l'Amazon GameLift Serversagente. Per tutti gli altri scenari di implementazione, chiama InitSDK() senza parametri.

Proprietà	Descrizione
webSocketUrl	I `GameLiftServerSdkEndpoint` Amazon GameLift Servers rendimenti quando si utilizza `RegisterCompute` una risorsa di calcolo Amazon GameLift Servers Anywhere. Tipo: `std::string`

Proprietà	Descrizione
	Campo obbligatorio: sì
ID del processo	Un identificatore univoco registrato nel processo del server che ospita il gioco. Tipo: `std::string` Campo obbligatorio: sì
hostId	`HostID`Viene `ComputeName` utilizzato al momento della registrazione del computer. Per ulteriori informazioni, consultare RegisterCompute. Tipo: `std::string` Campo obbligatorio: sì
FleetID	L'identificatore univoco del parco macchine su cui è registrato il computer. Per ulteriori informazioni, consultare RegisterCompute. Tipo: `std::string` Campo obbligatorio: sì
AuthToken	Il token di autenticazione generato da Amazon GameLift Servers questo autentica il server su. Amazon GameLift Servers Per ulteriori informazioni, consultare GetComputeAuthToken. Tipo: `std::string` Campo obbligatorio: sì

StartMatchBackfillRequest

Informazioni utilizzate per creare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni durante una chiamataAmazon GameLift Servers. StartMatchBackfill()

Proprietà	Descrizione
GameSessionArn	Un identificatore univoco della sessione di gioco. L'operazione API `GetGameSessionId` restituisce l'identificatore in formato ARN. Tipo: `std::string` Campo obbligatorio: sì
MatchmakingConfigurationArn	Un identificatore univoco, sotto forma di ARN, che il matchmaker può utilizzare per questa richiesta. L'ARN del matchmaker per la sessione di gioco originale si trova nell'oggetto della sessione di gioco nella proprietà matchmaker data. Scopri di più sui dati dei matchmaker in Work with matchmaker data. Tipo: `std::string` Campo obbligatorio: sì
Players	Un insieme di dati che rappresenta tutti i giocatori che partecipano alla sessione di gioco. Il matchmaker utilizza queste informazioni per cercare nuovi giocatori che rappresentano un buon abbinamento per i giocatori attuali. Tipo: `std::vector<Player>` Campo obbligatorio: sì
TicketId	Un identificatore univoco per un ticket di richiesta di matchmaking o match backfill.

Proprietà	Descrizione
	Se non fornisci un valore, Amazon GameLift Servers ne genera uno. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill degli abbinamenti o annullare la richiesta, se necessario. Tipo: `std::string` Required: No

Player

Questo tipo di dati rappresenta un giocatore in matchmaking. Quando si avvia una richiesta di matchmaking, un giocatore ha un ID giocatore, attributi e possibilmente dati di latenza. Amazon GameLift Serversaggiunge le informazioni sulla squadra dopo una partita.

Proprietà	Descrizione
LatencyInSM	Un insieme di valori espressi in milliseco ndi che indica la quantità di latenza che un giocatore sperimenta quando è connesso a una posizione. Se viene utilizzata questa proprietà, il giocatore viene abbinato solo per le posizioni elencate. Se un matchmaker ha una regola che valuta la latenza dei giocatori, i giocatori devono segnalare la latenza per essere abbinati. Tipo: `Dictionary<string,int>` Required: No
PlayerAttributes	Una raccolta di coppie chiave:valore contenent i informazioni sui giocatori da utilizzare nel matchmaking. Le chiavi degli attributi del giocatore devono corrispondere a quelle

Proprietà	Descrizione
	PlayerAttributes utilizzate in un set di regole di matchmaking. Per ulteriori informazioni sugli attributi dei giocatori, vedi AttributeValue. Tipo: `std::map<std::string,AttributeValue>` Required: No
PlayerId	Un identificatore univoco per un giocatore. Tipo: `std::string` Required: No
Team	Il nome della squadra a cui è assegnato il giocatore in una partita. Definisci il nome della squadra nel set di regole del matchmaking. Tipo: `std::string` Required: No

DescribePlayerSessionsRequest

Un oggetto che specifica le sessioni dei giocatori da recuperare. Il processo server fornisce queste informazioni con una DescribePlayerSessions() chiamata a. Amazon GameLift Servers

Proprietà	Descrizione
GameSessionId	Un identificatore univoco della sessione di gioco. Utilizzare questo parametro per richieder e tutte le sessioni giocatore per la sessione di gioco specificata.

Proprietà	Descrizione
	Il formato dell'ID della sessione di gioco è`arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>`. `GameSessionID` È una stringa ID personalizzata o un Tipo: `std::string` Required: No
PlayerSessionId	L'identificatore univoco per la sessione di un giocatore. Usa questo parametro per richiedere una singola sessione di gioco specifica. Tipo: `std::string` Required: No
PlayerId	L'identificatore univoco di un giocatore. Usa questo parametro per richiedere tutte le sessioni di gioco per un giocatore specifico. Consultare Genera giocatore IDs. Tipo: `std::string` Required: No

Proprietà	Descrizione
PlayerSessionStatusFilter	Lo stato della sessione del giocatore su cui filtrare i risultati. I possibili stati delle sessioni di gioco includono: • RISERVATO — La richiesta di sessione del giocatore è stata ricevuta, ma il giocatore non si è connesso al processo del server né è stato convalidato. • ATTIVO: il giocatore è stato convalidato dal processo del server ed è connesso. • COMPLETATA — La connessione del giocatore è stata interrotta. • TIMEDOUT: è stata ricevuta una richiesta di sessione, ma il giocatore non si è connesso o non è stato convalidato entro il limite di timeout (60 secondi). Tipo: `std::string` Required: No
NextToken	Il token che indica l'inizio della pagina successiva di risultati. Per specificare l'inizio del set di risultati, non fornire un valore. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `std::string` Required: No

Proprietà	Descrizione
Limite	Numero massimo di risultati da restituire. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `int` Required: No

StopMatchBackfillRequest

Informazioni utilizzate per annullare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni al Amazon GameLift Servers servizio durante una chiamata. StopMatchBackfill()

Proprietà	Descrizione
GameSessionArn	Un identificatore univoco della sessione di gioco della richiesta annullata. Tipo: `char[]` Required: No
MatchmakingConfigurationArn	Un identificatore univoco del matchmaker a cui è stata inviata questa richiesta. Tipo: `char[]` Required: No
TicketId	Un identificatore univoco del ticket di richiesta di riempimento da annullare. Tipo: `char[]` Required: No

AttributeValue

Utilizzate questi valori nelle Player coppie chiave-valore degli attributi. Questo oggetto consente di specificare il valore di un attributo utilizzando uno qualsiasi dei tipi di dati validi: stringa, numero, array di stringhe o mappa di dati. Ogni `AttributeValue` oggetto deve utilizzare esattamente una delle proprietà disponibili: `SN`,`SL`, o`SDM`.

Proprietà	Descrizione
AttrType	Speciifica il tipo di valore dell'attributo. I possibili tipi di valori degli attributi includono: • NONE • STRINGA • DOPPIA • ELENCO_STRINGHE • STRING_DOUBLE_MAP Required: No
S	Rappresenta un valore di attributo di tipo stringa. Tipo: `std::string` Required: No
N	Rappresenta un valore di attributo numerico. Tipo: `double` Required: No
SL	Rappresenta una matrice di valori di attributi di tipo stringa. Tipo: `std::vector<std::string>` Required: No
SDM	Rappresenta un dizionario di chiavi di stringa e valori doppi.

Proprietà	Descrizione
	Tipo: `std::map<std::string, double>` Required: No

GetFleetRoleCredentialsRequest

Questo tipo di dati offre al server di gioco un accesso limitato alle altre risorse. AWS Per ulteriori informazioni, consultare Configurare un ruolo di servizio IAM per Amazon GameLift Servers.

Proprietà	Descrizione
RoleArn	L'Amazon Resource Name (ARN) del ruolo di servizio che estende l'accesso limitato alle tue AWS risorse. Tipo: `std::string` Required: No
RoleSessionName	Il nome della sessione di ruolo che puoi utilizzare per identificare in modo univoco una AWS Security Token Service AssumeRolesessione. Questo nome è esposto nei log di controllo, come quelli di. CloudTrail Tipo: `std::string` Required: No

AwsLongOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione.

Proprietà	Descrizione
	Tipo: `long` Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `long&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

AwsStringOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: `std::string` Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `long&&`

Proprietà	Descrizione
	Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

DescribePlayerSessionsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “DescribePlayerSessionsResult” Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `Aws::GameLift::Server::Model::DescribePlayerSessionsResult&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì

Proprietà	Descrizione
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

DescribePlayerSessionsResult

Una raccolta di oggetti contenente proprietà per ogni sessione di giocatore che corrisponde alla richiesta.

Proprietà	Descrizione
NextToken	Un token che indica l'inizio della successiva pagina sequenzia le di risultati. Utilizzate il token restituito con una precedente chiamata a questa operazione. Per iniziare dall'inizio del set di risultati, non specificate un valore. Se viene specificato un ID sessione giocatore, questo parametro verrà ignorato. Tipo: `std::string` Campo obbligatorio: sì
PlayerSessions	Tipo: `IList<`the section called “PlayerSession” `>` Campo obbligatorio:
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `std::string&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool`

Proprietà	Descrizione
	Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

GenericOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

GenericOutcomeCallable

Questo tipo di dati è un risultato generico asincrono. Possiede le seguenti proprietà:

Proprietà	Descrizione
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì

Proprietà	Descrizione
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

PlayerSession

Questo tipo di dati rappresenta una sessione del giocatore che passa al server di gioco. Amazon GameLift Servers Per ulteriori informazioni, consulta PlayerSession.

Proprietà	Descrizione
CreationTime	Tipo: `long` Required: No
FleetId	Tipo: `std::string` Required: No
GameSessionId	Tipo: `std::string` Required: No
IpAddress	Tipo: `std::string` Required: No
PlayerData	Tipo: `std::string` Required: No
PlayerId	Tipo: `std::string` Required: No
PlayerSessionId	Tipo: `std::string`

Proprietà	Descrizione
	Required: No
Porta	Tipo: `int` Required: No
Stato	Stato sessione giocatore su cui filtrare i risultati. Quando PlayerId viene fornito un PlayerSessionId or, non PlayerSes sionStatusFilter ha alcun effetto sulla risposta. Tipo: Un `PlayerSessionStatus` enum. I valori possibili sono: • ACTIVE • COMPLETED • NOT_SET • RISERVATO • TIMEOUT Required: No
TerminationTime	Tipo: `long` Required: No
DnsName	Tipo: `std::string` Required: No

StartMatchBackfillOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione.

Proprietà	Descrizione
	Tipo: the section called “StartMatchBackfillResult” Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `StartMatchBackfillResult&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

StartMatchBackfillResult

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
TicketId	Un identificatore univoco per un ticket di matchmaking. Se qui non viene specificato alcun ID del ticket, ne Amazon GameLift Servers genererà uno sotto forma di UUID. Usa questo identific atore per tenere traccia dello stato del ticket di backfill della partita e recuperare i risultati della partita. Tipo: `std::string` Required: No

GetComputeCertificateOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “GetComputeCertificateResult” Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `Aws::GameLift::Server::Model::GetComputeCertificateResult&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

GetComputeCertificateResult

Il percorso del certificato TLS sul computer e il nome host del computer.

Proprietà	Descrizione
CertificatePath	Il percorso del certificato TLS sulla tua risorsa di elaborazione. Quando si utilizza una flotta Amazon GameLift Servers gestita, questo percorso contiene:

Proprietà	Descrizione
	• `certificate.pem` : Il certificato dell'utente finale. La catena completa di certificati è la combinazione dei certificati `certificateChain.pem` aggiunti a questo certificato. • `certificateChain.pem` : La catena di certificati che contiene il certificato principale e i certificati intermedi. • `rootCertificate.pem` : Il certificato principale. • `privateKey.pem` : la chiave privata per il certificato dell'utente finale. Tipo: `std::string` Required: No
ComputeName	Il nome della risorsa di calcolo. Tipo: `std::string` Required: No

GetFleetRoleCredentialsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “GetFleetRoleCredentialsResult” Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `Aws::GameLift::Server::Model::GetFleetRoleCredentialsResult`

Proprietà	Descrizione
	Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

GetFleetRoleCredentialsResult

Proprietà	Descrizione
AccessKeyId	L'ID della chiave di accesso per autenticare e fornire l'accesso alle tue risorse. AWS Tipo: `string` Required: No
AssumedRoleId	L'ID dell'utente a cui appartiene il ruolo di servizio. Tipo: `string` Required: No
AssumedRoleUserArn	L'Amazon Resource Name (ARN) dell'utente a cui appartiene il ruolo di servizio. Tipo: `string` Required: No

Proprietà	Descrizione
Scadenza	Il periodo di tempo che manca alla scadenza delle credenziali di sessione. Tipo: `DateTime` Required: No
SecretAccessKey	L'ID della chiave di accesso segreta per l'autenticazione. Tipo: `string` Required: No
SessionToken	Un token per identificare la sessione attiva corrente che interagisce con AWS le tue risorse. Tipo: `string` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

Inizialo SDKOutcome

> **Note**
>
> `InitSDKOutcome`viene restituito solo quando si crea l'SDK con il flag. `std` Se si crea con la `nostd` bandiera, allora the section called “GenericOutcome” viene invece restituito.

Proprietà	Descrizione
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

GameLiftError

Proprietà	Descrizione
ErrorType	Il tipo di errore. Tipo: Un enum. `GameLiftErrorType` Required: No
ErrorName	Il nome dell'errore. Tipo: `std::string` Required: No
ErrorMessage	Messaggio di errore. Tipo: `std::string` Required: No

enumerazioni;

Le enumerazioni definite per l'SDK del server per Amazon GameLift Servers (C++) sono definite come segue:

GameLiftErrorType

Valore di stringa che indica il tipo di errore. I valori validi includono:

- BAD_REQUEST_EXCEPTION
- GAMESESSION_ID_NOT_SET — L'ID della sessione di gioco non è stato impostato.
- INTERNAL_SERVICE_EXCEPTION
- LOCAL_CONNECTION_FAILED — La connessione locale a non è riuscita. Amazon GameLift Servers
- NETWORK_NOT_INITIALIZED — La rete non è stata inizializzata.
- SERVICE_CALL_FAILED — Una chiamata a un servizio non è riuscita. AWS
- WEBSOCKET_CONNECT_FAILURE
- WEBSOCKET_CONNECT_FAILURE_FORBIDDEN
- URL NON VALIDO DI WEBSOCKET_CONNECT_FAILURE_NON VALIDO
- WEBSOCKET_CONNECT_FAILURE_TIMEOUT
- ALREADY_INITIALIZED — Il server o il client è già stato inizializzato con Initialize (). Amazon GameLift Servers
- FLEET_MISMATCH — La flotta bersaglio non corrisponde alla flotta di una GameSession o PlayerSession.
- GAMELIFT_CLIENT_NOT_INITIALIZED — Il client non è stato inizializzato. Amazon GameLift Servers
- GAMELIFT_SERVER_NOT_INITIALIZED — Amazon GameLift Servers Il server non è stato inizializzato.
- GAME_SESSION_ENDED_FAILED: l'SDK del server non è riuscito a contattare il servizio per segnalare la fine della sessione di gioco. Amazon GameLift Servers
- GAME_SESSION_NOT_READY — La sessione di gioco sul server non è stata attivata. Amazon GameLift Servers
- GAME_SESSION_READY_FAILED: l'SDK del server non ha potuto contattare il servizio per segnalare che la sessione di gioco è pronta. Amazon GameLift Servers
- INITIALIZATION_MISMATCH — È stato chiamato un metodo client dopo Server: :Initialize () o viceversa.
- NOT_INITIALIZED — Il Amazon GameLift Servers server o il client non sono stati inizializzati con Initialize ().

- NO_TARGET_ALIASID_SET — Non è stato impostato un aliasID di destinazione.
- NO_TARGET_FLEET_SET — Non è stata impostata una flotta di obiettivi.
- PROCESS_ENDING_FAILED: l'SDK del server non è riuscito a contattare il servizio per segnalare la fine del processo. Amazon GameLift Servers
- PROCESS_NOT_ACTIVE — Il processo del server non è ancora attivo, non è associato a un e non può accettare o elaborare. GameSession PlayerSessions
- PROCESS_NOT_READY — Il processo del server non è ancora pronto per essere attivato.
- PROCESS_READY_FAILED: l'SDK del server non è riuscito a contattare il servizio per Amazon GameLift Servers segnalare che il processo è pronto.
- SDK_VERSION_DETECTION_FAILED — Rilevamento della versione SDK non riuscito.
- STX_CALL_FAILED — Una chiamata al componente di backend del server non è riuscita. XStx
- STX_INITIALIZATION_FAILED — L'inizializzazione del componente di backend del server non è riuscita. XStx
- UNEXPECTED_PLAYER_SESSION — Il server ha rilevato una sessione di giocatore non registrata.
- WEBSOCKET_CONNECT_FAILURE
- WEBSOCKET_CONNECT_FAILURE_FORBIDDEN
- URL NON VALIDO DI WEBSOCKET_CONNECT_FAILURE_NON VALIDO
- WEBSOCKET_CONNECT_FAILURE_TIMEOUT
- WEBSOCKET_RETRIABLE_SEND_MESSAGE_FAILURE — Errore recuperabile nell'invio di un messaggio al servizio. GameLift WebSocket
- WEBSOCKET_SEND_MESSAGE_FAILURE — Errore nell'invio di un messaggio al GameLift servizio. WebSocket
- MATCH_BACKFILL_REQUEST_VALIDATION — La convalida della richiesta non è riuscita.
- PLAYER_SESSION_REQUEST_VALIDATION — La convalida della richiesta non è riuscita.

PlayerSessionCreationPolicy

Valore della stringa che indica se la sessione di gioco accetta nuovi giocatori. I valori validi includono:

- ACCEPT_ALL: accetta tutte le nuove sessioni giocatore.
- DENY_ALL: rifiuta tutte le nuove sessioni giocatore.

- NOT_SET — La sessione di gioco non è impostata per accettare o rifiutare sessioni di nuovi giocatori.

Server C++ SDK 5.x per Amazon GameLift Servers -- Tipi di dati

Argomenti

GetSdkVersion()

Restituisce il numero di versione corrente dell'SDK integrato nel processo del server.

Sintassi

```
Aws::GameLift::AwsStringOutcome Server::GetSdkVersion();
```

Valore restituito

Se l'esito è positivo, restituisce la versione corrente dell'SDK come oggetto the section called "AwsStringOutcome". L'oggetto restituito include il numero di versione (esempio`5.0.0`). Se l'esito è negativo, verrà restituito un messaggio di errore.

Esempio

```
Aws::GameLift::AwsStringOutcome SdkVersionOutcome =
 Aws::GameLift::Server::GetSdkVersion();
```

InitSDK()

Inizializza l'SDK Amazon GameLift Servers. Chiama questo metodo all'avvio prima di qualsiasi altro passaggio di inizializzazione relativo aAmazon GameLift Servers. Questa azione legge i parametri del server dall'ambiente host per impostare la comunicazione tra il processo del server di gioco e il Amazon GameLift Servers servizio.

Se la build del server di gioco verrà implementata senza l'Amazon GameLift ServersAgent to a Amazon GameLift Servers Anywhere fleet o la flotta di container, chiama InitSDK() e specifica una serie di parametri del server.

Sintassi

```
Server::InitSDKOutcome Server::initSdkOutcome = InitSDK();
```

Valore restituito

Restituisce un the section called "Inizialo SDKOutcome" oggetto che indica se il processo del server è pronto per la chiamata. ProcessReady()

Esempio

```
//Call InitSDK to establish a local connection with the Amazon GameLift Servers Agent
 to enable further communication.
Aws::GameLift::Server::InitSDKOutcome initSdkOutcome =
  Aws::GameLift::Server::InitSDK();
```

InitSDK()

Inizializza l'SDK Amazon GameLift Servers. Chiama questo metodo all'avvio prima di qualsiasi altro passaggio di inizializzazione relativo aAmazon GameLift Servers. Questa azione richiede una serie di parametri del server per impostare la comunicazione tra il processo del server di gioco e il Amazon GameLift Servers servizio.

Se la build del server di gioco verrà distribuita a una EC2 flotta Amazon GameLift Servers gestita o a una flotta Amazon GameLift Servers Anywhere o a una flotta di container con l'Amazon GameLift ServersAgent, chiama InitSDK() senza parametri del server.

Sintassi

```
Server::InitSDKOutcome Server::initSdkOutcome = InitSDK(serverParameters);
```

Parametri

ServerParameters

Per inizializzare un server di gioco su una flotta Amazon GameLift Servers Anywhere, costruisci un `ServerParameters` oggetto con le seguenti informazioni:

- L'URL WebSocket utilizzato per connettersi al server di gioco.
- L'ID del processo utilizzato per ospitare il server di gioco.
- L'ID del computer che ospita i processi del server di gioco.
- L'ID della Amazon GameLift Servers flotta contenente il tuo computer Amazon GameLift Servers Anywhere.
- Il token di autorizzazione generato dall'Amazon GameLift Serversoperazione.

Valore restituito

Restituisce un the section called “Inizialo SDKOutcome” oggetto che indica se il processo del server è pronto per la chiamata. ProcessReady()

> **Note**
>
> Se le chiamate a non `InitSDK()` riescono per le build di gioco distribuite sulle flotte Anywhere, controlla il `ServerSdkVersion` parametro utilizzato durante la creazione della risorsa di compilazione. È necessario impostare esplicitamente questo valore sulla versione

SDK del server in uso. Il valore predefinito per questo parametro è 4.x, che non è compatibile. Per risolvere questo problema, crea una nuova build e distribuiscila in una nuova flotta.

Esempio

Amazon GameLift ServersEsempio: Ovunque

```
//Define the server parameters
std::string websocketUrl = "wss://us-west-1.api.amazongamelift.com";
std::string processId = "PID1234";
std::string fleetId = "arn:aws:gamelift:us-west-1:111122223333:fleet/
fleet-9999ffff-88ee-77dd-66cc-5555bbbb44aa";
std::string hostId = "HardwareAnywhere";
std::string authToken = "1111aaaa-22bb-33cc-44dd-5555eeee66ff";
Aws::GameLift::Server::Model::ServerParameters serverParameters =
  Aws::GameLift::Server::Model::ServerParameters(webSocketUrl, authToken, fleetId,
 hostId, processId);

//Call InitSDK to establish a local connection with the Amazon GameLift Servers Agent
 to enable further communication.
Aws::GameLift::Server::InitSDKOutcome initSdkOutcome =
 Aws::GameLift::Server::InitSDK(serverParameters);
```

ProcessReady()

Notifica Amazon GameLift Servers che il processo del server è pronto per ospitare sessioni di gioco. Chiama questo metodo dopo averlo InitSDK() invocato. Questo metodo deve essere chiamato solo una volta per processo.

Sintassi

```
GenericOutcome ProcessReady(const Aws::GameLift::Server::ProcessParameters
&processParameters);
```

Parametri

processParameters

Un oggetto ProcessParameters che comunica le informazioni seguenti sul processo del server:

- Nomi dei metodi di callback implementati nel codice del server di gioco richiamato dal Amazon GameLift Servers servizio per comunicare con il processo del server.

- Numero di porta sulla quale è in ascolto il processo del server.
- Percorso verso qualsiasi file specifico di una sessione di gioco che Amazon GameLift Servers deve acquisire e archiviare.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra la chiamata ProcessReady() e le implementazioni della funzione delegata.

```
// Set parameters and call ProcessReady
std::string serverLog("serverOut.log");        // Example of a log file written by the
 game server
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);
int listenPort = 9339;

Aws::GameLift::Server::ProcessParameters processReadyParameter =
  Aws::GameLift::Server::ProcessParameters(
    std::bind(&Server::onStartGameSession, this, std::placeholders::_1),
    std::bind(&Server::onProcessTerminate, this),
    std::bind(&Server::OnHealthCheck, this),
    std::bind(&Server::OnUpdateGameSession, this),
    listenPort,
    Aws::GameLift::Server::LogParameters(logPaths)
  );

Aws::GameLift::GenericOutcome outcome =
  Aws::GameLift::Server::ProcessReady(processReadyParameter);

// Implement callback functions
void Server::onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
  // game-specific tasks when starting a new game session, such as loading map
  GenericOutcome outcome =
    Aws::GameLift::Server::ActivateGameSession (maxPlayers);
}

void Server::onProcessTerminate()
{
```

```
  // game-specific tasks required to gracefully shut down a game session,
  // such as notifying players, preserving game state data, and other cleanup
  GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
}

bool Server::onHealthCheck()
{
  bool health;
  // complete health evaluation within 60 seconds and set health
  return health;
}
```

ProcessReadyAsync()

Notifica al servizio Amazon GameLift Servers che il processo del server è pronto per l'hosting delle sessioni di gioco. Questo metodo dovrebbe essere chiamato dopo che il processo del server è pronto per ospitare una sessione di gioco. I parametri specificano i nomi delle funzioni di callback Amazon GameLift Servers da chiamare in determinate circostanze. Il codice del server di gioco deve implementare queste funzioni.

Questa chiamata è asincrona. Per effettuare una chiamata sincrona, utilizza ProcessReady(). Per ulteriori dettagli, consulta Inizializza il processo del server.

Sintassi

```
GenericOutcomeCallable ProcessReadyAsync(
    const Aws::GameLift::Server::ProcessParameters &processParameters);
```

Parametri

processParameters

Un oggetto ProcessParameters che comunica le informazioni seguenti sul processo del server:

- Nomi dei metodi di callback implementati nel codice del server di gioco richiamato dal Amazon GameLift Servers servizio per comunicare con il processo del server.
- Numero di porta sulla quale è in ascolto il processo del server.
- Percorso verso qualsiasi file specifico di una sessione di gioco che Amazon GameLift Servers deve acquisire e archiviare.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
// Set parameters and call ProcessReady
std::string serverLog("serverOut.log");        // This is an example of a log file
 written by the game server
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);
int listenPort = 9339;

Aws::GameLift::Server::ProcessParameters processReadyParameter =
  Aws::GameLift::Server::ProcessParameters(std::bind(&Server::onStartGameSession, this,
 std::placeholders::_1),
  std::bind(&Server::onProcessTerminate, this), std::bind(&Server::OnHealthCheck,
 this),
  std::bind(&Server::OnUpdateGameSession, this), listenPort,
 Aws::GameLift::Server::LogParameters(logPaths));

Aws::GameLift::GenericOutcomeCallable outcome =
  Aws::GameLift::Server::ProcessReadyAsync(processReadyParameter);

// Implement callback functions
void onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
  // game-specific tasks when starting a new game session, such as loading map
  GenericOutcome outcome = Aws::GameLift::Server::ActivateGameSession (maxPlayers);
}

void onProcessTerminate()
{
  // game-specific tasks required to gracefully shut down a game session,
  // such as notifying players, preserving game state data, and other cleanup
  GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
}

bool onHealthCheck()
{
  // perform health evaluation and complete within 60 seconds
  return health;
}
```

ProcessEnding()

Notifica Amazon GameLift Servers che il processo del server sta terminando. Chiama questo metodo dopo tutte le altre attività di pulizia (inclusa la chiusura della sessione di gioco attiva) e prima di terminare il processo. A seconda del risultato di`ProcessEnding()`, il processo termina con successo (0) o errore (-1) e genera un evento relativo alla flotta. Se il processo termina con un errore, l'evento fleet generato è. `SERVER_PROCESS_TERMINATED_UNHEALTHY`

Sintassi

```
Aws::GameLift::GenericOutcome processEndingOutcome =
 Aws::GameLift::Server::ProcessEnding();
```

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio chiama `ProcessEnding()` e `Destroy()` prima di terminare il processo del server con un codice di uscita riuscito o di errore.

```
Aws::GameLift::GenericOutcome processEndingOutcome =
 Aws::GameLift::Server::ProcessEnding();
Aws::GameLift::Server::Destroy();

// Exit the process with success or failure
if (processEndingOutcome.IsSuccess()) {
  exit(0);
}
else {
  cout << "ProcessEnding() failed. Error: " <<
 processEndingOutcome.GetError().GetErrorMessage();
  exit(-1);
}
```

ActivateGameSession()

Notifica Amazon GameLift Servers che il processo del server ha attivato una sessione di gioco ed è ora pronto a ricevere le connessioni dei giocatori. Questa azione dovrebbe essere richiamata come parte della funzione di on`StartGameSession()` callback, dopo l'inizializzazione di tutta la sessione di gioco.

Sintassi

```
Aws::GameLift::GenericOutcome activateGameSessionOutcome =
 Aws::GameLift::Server::ActivateGameSession();
```

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio mostra le `ActivateGameSession()` chiamate come parte della funzione `onStartGameSession()` delegata.

```
void onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
  // game-specific tasks when starting a new game session, such as loading map
  GenericOutcome outcome = Aws::GameLift::Server::ActivateGameSession();
}
```

UpdatePlayerSessionCreationPolicy()

Aggiorna la capacità della sessione di gioco corrente di accettare nuove sessioni giocatore. Una sessione di gioco può essere configurata per accettare o rifiutare tutte le nuove sessioni giocatore.

Sintassi

```
GenericOutcome
 UpdatePlayerSessionCreationPolicy(Aws::GameLift::Model::PlayerSessionCreationPolicy
 newPlayerSessionPolicy);
```

Parametri

playerCreationSessionPolitica

Tipo: valore `PlayerSessionCreationPolicy` enum.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio definisce la policy di partecipazione alla sessione di gioco corrente per accettare tutti i giocatori.

```
Aws::GameLift::GenericOutcome outcome =

 Aws::GameLift::Server::UpdatePlayerSessionCreationPolicy(Aws::GameLift::Model::PlayerSessionCr
```

GetGameSessionId()

Recupera l'ID della sessione di gioco ospitata dal processo del server attivo.

Per i processi inattivi che non vengono attivati con una sessione di gioco, la chiamata restituisce un. the section called “GameLiftError”

Sintassi

```
AwsStringOutcome GetGameSessionId()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Se l'esito è positivo, l'ID della sessione di gioco verrà restituito come oggetto the section called “AwsStringOutcome”. Se l'esito è negativo, verrà restituito un messaggio di errore.

Per i processi inattivi che non vengono attivati con una sessione di gioco, la chiamata restituisce `Success` = `True` e `GameSessionId` =. `""`

Esempio

```
Aws::GameLift::AwsStringOutcome sessionIdOutcome =
 Aws::GameLift::Server::GetGameSessionId();
```

GetTerminationTime()

Restituisce il tempo di arresto pianificato di un processo del server, se è disponibile un tempo di chiusura. Un processo server interviene dopo aver ricevuto una `onProcessTerminate()`

richiamata da. Amazon GameLift Servers Amazon GameLift Serverschiama `onProcessTerminate()` per i seguenti motivi:

- Quando il processo del server ha segnalato problemi di salute o non ha risposto. Amazon GameLift Servers
- Quando si termina l'istanza durante un evento di scale-down.
- Quando un'istanza viene terminata a causa di un'interruzione di un'istanza spot.

Sintassi

```
AwsDateTimeOutcome GetTerminationTime()
```

Valore restituito

In caso di successo, restituisce l'ora di terminazione come oggetto. `AwsDateTimeOutcome` Il valore è il tempo di terminazione, espresso in segni di spunta trascorsi da allora. `0001 00:00:00` Ad esempio, il valore della data e dell'ora è uguale ai segni di `2020-09-13 12:26:40 -000Z` spunta. `637355968000000000` Se non è disponibile alcun orario di terminazione, restituisce un messaggio di errore.

Se il processo non ha ricevuto un ProcessParameters. OnProcessTerminate() callback, viene restituito un messaggio di errore. Per ulteriori informazioni sulla chiusura di un processo server, vedere. Rispondi a una notifica di chiusura di un processo del server

Esempio

```
Aws::GameLift::AwsLongOutcome TermTimeOutcome =
 Aws::GameLift::Server::GetTerminationTime();
```

AcceptPlayerSession()

Notifica Amazon GameLift Servers che un giocatore con l'ID di sessione specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift Serversverifica che l'ID della sessione del giocatore sia valido. Dopo la convalida della sessione di gioco, Amazon GameLift Servers modifica lo stato dello slot del giocatore da RISERVATO ad ATTIVO.

Sintassi

```
GenericOutcome AcceptPlayerSession(String playerSessionId)
```

Parametri

playerSessionId

ID univoco rilasciato Amazon GameLift Servers quando viene creata una nuova sessione giocatore.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio gestisce una richiesta di connessione che include la convalida e il rifiuto di una sessione giocatore non valida. IDs

```
void ReceiveConnectingPlayerSessionID (Connection& connection, const std::string&
 playerSessionId)
{
  Aws::GameLift::GenericOutcome connectOutcome =
 Aws::GameLift::Server::AcceptPlayerSession(playerSessionId);
  if(connectOutcome.IsSuccess())
  {
    connectionToSessionMap.emplace(connection, playerSessionId);
    connection.Accept();
  }
  else
  {
    connection.Reject(connectOutcome.GetError().GetMessage();
  }
}
```

RemovePlayerSession()

Notifica Amazon GameLift Servers che un giocatore si è disconnesso dal processo del server. In risposta, Amazon GameLift Servers modifica lo slot del giocatore rendendolo disponibile.

Sintassi

```
GenericOutcome RemovePlayerSession(String playerSessionId)
```

Parametri

`playerSessionId`

ID univoco rilasciato Amazon GameLift Servers quando viene creata una nuova sessione di gioco.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
Aws::GameLift::GenericOutcome disconnectOutcome =
 Aws::GameLift::Server::RemovePlayerSession(playerSessionId);
```

DescribePlayerSessions()

Recupera i dati della sessione del giocatore che includono impostazioni, metadati della sessione e dati del giocatore. Utilizzate questo metodo per ottenere informazioni su quanto segue:

- Una sessione per giocatore singolo
- Tutte le sessioni dei giocatori in una sessione di gioco
- Tutte le sessioni dei giocatori sono associate a un ID giocatore singolo

Sintassi

```
DescribePlayerSessionsOutcome DescribePlayerSessions(DescribePlayerSessionsRequest
 describePlayerSessionsRequest)
```

Parametri

DescribePlayerSessionsRequest

Un the section called “DescribePlayerSessionsRequest” oggetto che descrive le sessioni dei giocatori da recuperare.

Valore restituito

Se l'esito è positivo, restituisce un oggetto the section called “DescribePlayerSessionsOutcome” contenente un set di oggetti di sessione giocatore corrispondente ai parametri della richiesta.

Esempio

Questo esempio richiede che tutte le sessioni dei giocatori siano connesse attivamente a una sessione di gioco specificata. Omettendo NextTokene impostando il valore Limite su 10, Amazon GameLift Servers restituisce i primi 10 record di sessione tra giocatori corrispondenti alla richiesta.

```
// Set request parameters
Aws::GameLift::Server::Model::DescribePlayerSessionsRequest request;
request.SetPlayerSessionStatusFilter(Aws::GameLift::Server::Model::PlayerSessionStatusMapper::G
request.SetLimit(10);
request.SetGameSessionId("the game session ID");    // can use GetGameSessionId()

// Call DescribePlayerSessions
Aws::GameLift::DescribePlayerSessionsOutcome playerSessionsOutcome =
 Aws::GameLift::Server::DescribePlayerSessions(request);
```

StartMatchBackfill()

Invia una richiesta per trovare nuovi giocatori per gli slot aperti in una sessione di gioco creata con FlexMatch. Per ulteriori informazioni, consulta la funzione di FlexMatchriempimento.

Questa operazione è asincrona. Se vengono abbinati nuovi giocatori, Amazon GameLift Servers fornisce dati aggiornati sul matchmaker utilizzando la funzione di callback. `OnUpdateGameSession()`

Un processo del server può avere un solo backfill degli abbinamenti attivo alla volta. Per inviare una nuova richiesta, chiama prima StopMatchBackfill() per annullare la richiesta originale.

Sintassi

```
StartMatchBackfillOutcome StartMatchBackfill (StartMatchBackfillRequest
 startBackfillRequest);
```

Parametri

StartMatchBackfillRequest

Un StartMatchBackfillRequest oggetto che comunica le seguenti informazioni:

- ID del ticket da assegnare alla richiesta di backfill. Queste informazioni sono facoltative; se non viene fornito alcun ID, ne Amazon GameLift Servers genererà uno.

- Matchmaker a cui inviare la richiesta. L'ARN di configurazione completo è obbligatorio. Questo valore si trova nei dati del matchmaker della sessione di gioco.
- L'ID della sessione di gioco da riempire.
- I dati di matchmaking disponibili per i giocatori attuali della sessione di gioco.

Valore restituito

Restituisce un the section called “StartMatchBackfillOutcome” oggetto con l'ID del ticket match backfill, oppure restituisce un errore con un messaggio di errore.

Esempio

```
// Build a backfill request
std::vector<Player> players;
Aws::GameLift::Server::Model::StartMatchBackfillRequest startBackfillRequest;
startBackfillRequest.SetTicketId("1111aaaa-22bb-33cc-44dd-5555eeee66ff");  // optional,
 autogenerated if not provided
startBackfillRequest.SetMatchmakingConfigurationArn("arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig"); //from the game
 session matchmaker data
startBackfillRequest.SetGameSessionArn("the game session ARN");        // can use
 GetGameSessionId()
startBackfillRequest.SetPlayers(players);                                  // from the
 game session matchmaker data

// Send backfill request
Aws::GameLift::StartMatchBackfillOutcome backfillOutcome =
  Aws::GameLift::Server::StartMatchBackfill(startBackfillRequest);

// Implement callback function for backfill
void Server::OnUpdateGameSession(Aws::GameLift::Server::Model::GameSession gameSession,
 Aws::GameLift::Server::Model::UpdateReason updateReason, std::string backfillTicketId)
{
  // handle status messages
  // perform game-specific tasks to prep for newly matched players
}
```

StopMatchBackfill()

Annulla una richiesta di match backfill attiva. Per ulteriori informazioni, consulta la funzione di FlexMatchriempimento.

Sintassi

```
GenericOutcome StopMatchBackfill (StopMatchBackfillRequest stopBackfillRequest);
```

Parametri

StopMatchBackfillRequest

Un StopMatchBackfillRequest oggetto che identifica il ticket di matchmaking da annullare:

- L'ID del ticket assegnato alla richiesta di riempimento.
- Il matchmaker a cui è stata inviata la richiesta di riempimento.
- La sessione di gioco associata alla richiesta di riempimento.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
// Set backfill stop request parameters

Aws::GameLift::Server::Model::StopMatchBackfillRequest stopBackfillRequest;
stopBackfillRequest.SetTicketId("1111aaaa-22bb-33cc-44dd-5555eeee66ff");
stopBackfillRequest.SetGameSessionArn("the game session ARN"); // can use
 GetGameSessionId()
stopBackfillRequest.SetMatchmakingConfigurationArn("arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig");
// from the game session matchmaker data

Aws::GameLift::GenericOutcome stopBackfillOutcome =
  Aws::GameLift::Server::StopMatchBackfill(stopBackfillRequest);
```

GetComputeCertificate()

Recupera il percorso del certificato TLS utilizzato per crittografare la connessione di rete tra la risorsa di elaborazione Amazon GameLift Servers Anywhere e. Amazon GameLift Servers Puoi utilizzare il percorso del certificato quando registri il tuo dispositivo di elaborazione in una flotta Anywhere. Amazon GameLift Servers Per ulteriori informazioni, consultare RegisterCompute.

Sintassi

```
GetComputeCertificateOutcome Server::GetComputeCertificate()
```

Valore restituito

Restituisce una the section called “GetComputeCertificateOutcome”.

Esempio

```
Aws::GameLift::GetComputeCertificateOutcome certificate =
 Aws::GameLift::Server::GetComputeCertificate();
```

GetFleetRoleCredentials()

Recupera le credenziali del ruolo IAM che autorizzano Amazon GameLift Servers a interagire con altri. Servizi AWS Per ulteriori informazioni, consulta Comunica con altre AWS risorse delle tue flotte.

Sintassi

```
GetFleetRoleCredentialsOutcome GetFleetRoleCredentials(GetFleetRoleCredentialsRequest
 request);
```

Parametri

GetFleetRoleCredentialsRequest

Valore restituito

Restituisce un oggetto the section called “GetFleetRoleCredentialsOutcome”.

Esempio

```
// form the fleet credentials request
Aws::GameLift::Server::Model::GetFleetRoleCredentialsRequest
 getFleetRoleCredentialsRequest;
getFleetRoleCredentialsRequest.SetRoleArn("arn:aws:iam::123456789012:role/service-role/
exampleGameLiftAction");

Aws::GameLift::GetFleetRoleCredentialsOutcome credentials =
 Aws::GameLift::Server::GetFleetRoleCredentials(getFleetRoleCredentialsRequest);
```

Questo esempio mostra l'uso del `RoleSessionName` valore opzionale per assegnare un nome alla sessione di credenziali a fini di controllo. Se non si fornisce un nome per la sessione di ruolo, viene utilizzato il valore predefinito *`[host-id]`* "*`[fleet-id]`*-».

```
// form the fleet credentials request
Aws::GameLift::Server::Model::GetFleetRoleCredentialsRequest
 getFleetRoleCredentialsRequest;
getFleetRoleCredentialsRequest.SetRoleArn("arn:aws:iam::123456789012:role/service-role/
exampleGameLiftAction");
getFleetRoleCredentialsRequest.SetRoleSessionName("MyFleetRoleSession");

Aws::GameLift::GetFleetRoleCredentialsOutcome credentials =
 Aws::GameLift::Server::GetFleetRoleCredentials(getFleetRoleCredentialsRequest);
```

Distruggi ()

Libera l'SDK del server di Amazon GameLift Servers gioco dalla memoria. È consigliabile chiamare questo metodo dopo `ProcessEnding()` e prima di terminare il processo. Se utilizzi una flotta Anywhere e non interrompi i processi del server dopo ogni sessione di gioco, chiama `Destroy()` e poi `InitSDK()` reinizializza prima di notificare Amazon GameLift Servers che il processo è pronto per ospitare una sessione di gioco. `ProcessReady()`

Sintassi

```
GenericOutcome Aws::GameLift::Server::Destroy();
```

Parametri

Non ci sono parametri.

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

```
Aws::GameLift::GenericOutcome processEndingOutcome =
 Aws::GameLift::Server::ProcessEnding();
Aws::GameLift::Server::Destroy();

// Exit the process with success or failure
if (processEndingOutcome.IsSuccess()) {
```

```
  exit(0);
}
else {
  cout << "ProcessEnding() failed. Error: " <<
 processEndingOutcome.GetError().GetErrorMessage();
  exit(-1);
}
```

Server C# SDK 5.x per -- Azioni Amazon GameLift Servers

Usa il riferimento all'SDK 5.x del server per integrare il tuo gioco multiplayer con cui ospitare. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consulta. Aggiungi Amazon GameLift Servers al tuo server di gioco Se stai usando il Amazon GameLift Servers plugin per Unity, vedi ancheAmazon GameLift Serversplugin per Unity (server SDK 5.x).

Server C# SDK 5.x per Amazon GameLift Servers -- Tipi di dati

Usa il riferimento all'SDK 5.x del server Amazon GameLift Servers C# per integrare il tuo gioco multiplayer con cui ospitare. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consulta. Aggiungi Amazon GameLift Servers al tuo server di gioco Se stai usando il Amazon GameLift Servers plugin per Unity, vedi ancheAmazon GameLift Serversplugin per Unity (server SDK 5.x).

Server C# SDK 5.x per -- Azioni Amazon GameLift Servers

Tipi di dati

- LogParameters
- ProcessParameters
- UpdateGameSession
- GameSession
- ServerParameters
- StartMatchBackfillRequest
- Player
- DescribePlayerSessionsRequest
- StopMatchBackfillRequest
- GetFleetRoleCredentialsRequest
- AttributeValue

- AwsStringOutcome
- GenericOutcome
- DescribePlayerSessionsOutcome
- DescribePlayerSessionsResult
- PlayerSession
- StartMatchBackfillOutcome
- StartMatchBackfillResult
- GetComputeCertificateOutcome
- GetComputeCertificateResult
- GetFleetRoleCredentialsOutcome
- GetFleetRoleCredentialsResult
- AwsDateTimeOutcome
- GameLiftError
- enumerazioni;

LogParameters

Usa questo tipo di dati per identificare i file generati durante una sessione di gioco su cui desideri che il server di gioco venga caricato al Amazon GameLift Servers termine della sessione di gioco. Il server di gioco comunica `LogParameters to` Amazon GameLift Servers durante una ProcessReady() chiamata.

Proprietà	Descrizione
LogPaths	L'elenco dei percorsi di directory dei file di registro del server di gioco che desideri Amazon GameLift Servers archiviare per accessi futuri. Il processo del server genera questi file durante ogni sessione di gioco. Definisci i percorsi e i nomi dei file nel tuo server di gioco e li memorizzi nella directory di build del gioco principale.

	I percorsi di registro devono essere assoluti. Ad esempio, se la build del gioco memorizza i log delle sessioni di gioco in un percorso simile`MyGame\sessionLogs\` , il percorso si troverebbe `c:\game\MyGame\sessionLogs` su un'istanza di Windows. Tipo: `List<String>` Required: No

ProcessParameters

Questo tipo di dati contiene l'insieme di parametri Amazon GameLift Servers a cui viene inviato una ProcessReady() chiamata.

Proprietà	Descrizione
LogParameters	L'oggetto con un elenco di percorsi di directory ai file di registro delle sessioni di gioco. Tipo: `Aws::GameLift::Server::` LogParameters Campo obbligatorio: sì
OnHealthCheck	Il nome della funzione di callback che Amazon GameLift Servers richiama per richiedere un rapporto sullo stato di salute del processo del server. Amazon GameLift Serverschiama questa funzione ogni 60 secondi. Dopo aver chiamato questa funzione Amazon GameLift Servers attende 60 secondi per una risposta, se non ne riceve nessuna, Amazon GameLift Servers registra il processo del server come non integro. Tipo: `void OnHealthCheckDelegate()`

	Campo obbligatorio: sì
OnProcessTerminate	Il nome della funzione di callback che Amazon GameLift Servers richiama per forzare l'arresto del processo server. Dopo aver richiamato questa funzione, Amazon GameLift Servers attende per cinque minuti l'arresto del processo del server e risponde con una chiamata ProcessEnding() prima di arrestare il processo del server. Tipo: `void OnProcessTerminate Delegate()` Campo obbligatorio: sì
OnStartGameSession	Il nome della funzione di callback che Amazon GameLift Servers richiama per attivare una nuova sessione di gioco. Amazon GameLift Serverschiama questa funzione in risposta alla richiesta del client. CreateGameSession La funzione di callback accetta un GameSession oggetto. Tipo: `void OnStartGameSession Delegate( GameSession )` Campo obbligatorio: sì

OnUpdateGameSession	Il nome della funzione di callback che Amazon GameLift Servers richiama per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift Serverschiama questa funzione quando è stata elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un GameSession oggetto, uno status update (`updateReason`) e l'ID del ticket match backfill. Tipo: void () OnUpdateGameSessionDelegate UpdateGameSession Required: No
Porta	Il numero di porta su cui il processo server ascolta le connessioni di nuovi giocatori . Il valore deve rientrare nella gamma di porte configurate per qualsiasi parco istanze che distribuisce questa build del server di gioco. Questo numero di porta è incluso nella sessione di gioco e negli oggetti della sessione del giocatore utilizzati dalle sessioni di gioco per la connessione a un processo server. Tipo: `Integer` Campo obbligatorio: sì

UpdateGameSession

Le informazioni aggiornate relative a un oggetto della sessione di gioco includono il motivo per cui la sessione di gioco è stata aggiornata. Se l'aggiornamento è correlato a un'azione di ripristino di una partita, questo tipo di dati include l'ID del ticket di backfill.

Proprietà	Descrizione
GameSession	Oggetto GameSession. L'`GameSessi on` oggetto contiene proprietà che descrivono una sessione di gioco. Tipo: `GameSession GameSession()` Campo obbligatorio: sì
UpdateReason	Il motivo per cui la sessione di gioco viene aggiornata. Tipo: `UpdateReason UpdateReason()` Campo obbligatorio: sì
BackfillTicketId	L'ID del ticket di backfill che tenta di aggiornare la sessione di gioco. Tipo: `String` Campo obbligatorio: sì

GameSession

Dettagli di una sessione di gioco.

Proprietà	Descrizione
GameSessionId	Un identificatore univoco per la sessione di gioco. L'ARN di una sessione di gioco ha il seguente formato:. `arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token>` Tipo: `String`

Proprietà	Descrizione
	Required: No
Nome	Un'etichetta descrittiva della sessione di gioco. Tipo: `String` Required: No
FleetId	Un identificatore univoco per la flotta su cui è in esecuzione la sessione di gioco. Tipo: `String` Required: No
MaximumPlayerSessionCount	Il numero massimo di connessioni dei giocatori alla sessione di gioco. Tipo: `Integer` Required: No
Porta	Il numero di porta per la sessione di gioco. Per connettersi a un server di Amazon GameLift Servers gioco, un'app necessita sia dell'indi rizzo IP che del numero di porta. Tipo: `Integer` Required: No
IpAddress	L'indirizzo IP della sessione di gioco. Per connettersi a un server di Amazon GameLift Servers gioco, un'app necessita sia dell'indi rizzo IP che del numero di porta. Tipo: `String` Required: No

Proprietà	Descrizione
GameSessionData	Un set di proprietà personalizzate della sessione di gioco, formattate come valore di stringa singola. Tipo: `String` Required: No
MatchmakerData	Le informazioni sul processo di matchmaking utilizzato per creare la sessione di gioco, in sintassi JSON, sono formattate come stringa. Oltre alla configurazione di matchmaking utilizzata, contiene dati su tutti i giocatori assegnati alla partita, compresi gli attributi dei giocatori e le assegnazioni delle squadre. Tipo: `String` Required: No
GameProperties	Un insieme di proprietà personalizzate per una sessione di gioco, formattate come coppie chiave:valore. Queste proprietà vengono passate con una richiesta di avvio di una nuova sessione di gioco. Tipo: `Dictionary<string, string>` Required: No

Proprietà	Descrizione
DnsName	L'identificatore DNS assegnato all'istanza che esegue la sessione di gioco. I valori hanno il seguente formato: • Flotte abilitate per TLS:. `<unique identifier>.<region identifier>.amazongamelift.com` • Non-TLS-enabled flotte:. `ec2-<unique identifier>.compute.amazonaws.com` Quando ti connetti a una sessione di gioco in esecuzione su una flotta abilitata per TLS, devi usare il nome DNS, non l'indirizzo IP. Tipo: `String` Required: No

ServerParameters

Informazioni utilizzate per mantenere la connessione tra un server Amazon GameLift Servers Anywhere e il servizio. Amazon GameLift Servers Queste informazioni vengono utilizzate quando si avviano nuovi processi server conInitSDK(). Per i server ospitati su EC2 istanze Amazon GameLift Servers gestite, utilizza un oggetto vuoto.

Proprietà	Descrizione
WebSocketUrl	I `GameLiftServerSdkEndpoint` restituiti quando fai parte `RegisterCompute` di Amazon GameLift Servers Anywhere. Tipo: `String` Campo obbligatorio: sì

Proprietà	Descrizione
ProcessId	Un identificatore univoco registrato nel processo del server che ospita il gioco. Tipo: `String` Campo obbligatorio: sì
HostId	Un identificatore univoco per l'host con i processi del server che ospitano il gioco. L'HostID viene ComputeName utilizzato quando hai registrato il tuo computer. Per ulteriori informazioni, consulta, RegisterCompute Tipo: `String` Campo obbligatorio: sì
FleetId	L'ID del parco veicoli su cui è registrato il computer. Per ulteriori informazioni, consultare RegisterCompute. Tipo: `String` Campo obbligatorio: sì
AuthToken	Il token di autenticazione generato da Amazon GameLift Servers questo codice autentica il server. Amazon GameLift Servers Per ulteriori informazioni, consultare GetComputeAuthToken. Tipo: `String` Campo obbligatorio: sì

StartMatchBackfillRequest

Informazioni utilizzate per creare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni durante una chiamataAmazon GameLift Servers. StartMatchBackfill()

Proprietà	Descrizione
GameSessionArn	L'identificatore univoco della sessione di gioco. L'operazione API `GetGameSessionId` restituisce l'identificatore in formato ARN. Tipo: `String` Campo obbligatorio: sì
MatchmakingConfigurationArn	L'identificatore univoco, sotto forma di ARN, che il matchmaker deve utilizzare per questa richiesta. L'ARN del matchmaker per la sessione di gioco originale si trova nell'ogge tto della sessione di gioco nella proprietà matchmaker data. Scopri di più sui dati dei matchmaker in Work with matchmaker data. Tipo: `String` Campo obbligatorio: sì
Players	Un insieme di dati che rappresenta tutti i giocatori attualmente impegnati nella sessione di gioco. Il matchmaker utilizza queste informazioni per cercare nuovi giocatori che rappresentano un buon abbinamento per i giocatori attuali. Tipo: `List<Player>` Campo obbligatorio: sì
TicketId	L'identificatore univoco di un ticket di richiesta di matchmaking o match backfill. Se non

Proprietà	Descrizione
	fornisci un valore, Amazon GameLift Servers ne genera uno. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill degli abbinamenti o annullare la richiesta, se necessario. Tipo: `String` Required: No

Player

Rappresenta un giocatore in matchmaking. Quando inizia una richiesta di matchmaking, un giocatore ha un ID giocatore, attributi e possibilmente dati sulla latenza. Amazon GameLift Serversaggiunge le informazioni sulla squadra dopo una partita.

Proprietà	Descrizione
LatencyInSM	Un insieme di valori espressi in milliseco ndi, che indica la quantità di latenza che un giocatore sperimenta quando è connesso a una posizione. Se viene utilizzata questa proprietà, il giocatore viene abbinato solo per le posizioni elencate. Se un matchmaker ha una regola che valuta la latenza dei giocatori, i giocatori devono segnalare la latenza per essere abbinati. Tipo: `Dictionary<string, int>` Required: No
PlayerAttributes	Una raccolta di coppie chiave:valore che contengono informazioni sui giocatori da utilizzare nel matchmaking. Le chiavi degli attributi del giocatore devono corrispondere a

Proprietà	Descrizione
	quelle PlayerAttributes utilizzate in un set di regole di matchmaking. Per ulteriori informazioni sugli attributi dei giocatori, vedi AttributeValue. Tipo: `Dictionary<string, AttributeValue` Required: No
PlayerId	Un identificatore univoco per un giocatore. Tipo: `String` Required: No
Team	Il nome della squadra a cui è assegnato il giocatore in una partita. Definisci il nome della squadra nel set di regole del matchmaking. Tipo: `String` Required: No

DescribePlayerSessionsRequest

Questo tipo di dati viene utilizzato per specificare quale sessione del giocatore recuperare. Può essere utilizzato in diversi modi: (1) fornire PlayerSessionId a per richiedere una sessione di gioco specifica; (2) fornire GameSessionId a richiedere tutte le sessioni di gioco nella sessione di gioco specificata; o (3) fornire PlayerId a per richiedere tutte le sessioni di gioco per il giocatore specificato. Per le raccolte di grandi dimensioni delle sessioni giocatore, utilizzare i parametri di paginazione per recuperare i risultati in pagine sequenziali.

Proprietà	Descrizione
GameSessionId	L'identificatore unico della sessione di gioco. Utilizzare questo parametro per richiedere

Proprietà	Descrizione
	tutte le sessioni giocatore per la sessione di gioco specificata. Il formato dell'ID della sessione di gioco è il seguente: `arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>` . Il valore di <ID string> corrisponde a una stringa ID personalizzata o (se ne è stata specificata una al momento della creazione della sessione di gioco) a una stringa generata. Tipo: `String` Required: No
PlayerSessionId	L'identificatore univoco per la sessione di un giocatore. Tipo: `String` Required: No
PlayerId	L'identificatore univoco di un giocatore. Consultare Genera giocatore IDs. Tipo: `String` Required: No

Proprietà	Descrizione
PlayerSessionStatusFilter	Lo stato della sessione del giocatore su cui filtrare i risultati. Tra gli stati sessione giocatore possibili sono inclusi i seguenti: • RISERVATO — La richiesta di sessione del giocatore è stata ricevuta, ma il giocatore non si è ancora connesso al server. La procedura and/or è stata convalidata. • ACTIVE (ATTIVO) - Il giocatore è stato convalidato dal processo del server ed è attualmente collegato. • COMPLETED (COMPLETATO) - La connessione del giocatore è stata interrotta. • TIMEDOUT — È stata ricevuta una richiesta di sessione di gioco, ma il giocatore che non si and/or è connesso non è stata convalidata entro il limite di timeout (60 secondi). Tipo: `String` Required: No
NextToken	Il token che indica l'inizio della pagina successiva di risultati. Per specificare l'inizio del set di risultati, non fornire un valore. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `String` Required: No

Proprietà	Descrizione
Limite	Numero massimo di risultati da restituire. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `int` Required: No

StopMatchBackfillRequest

Informazioni utilizzate per annullare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni al Amazon GameLift Servers servizio durante una chiamata. StopMatchBackfill()

Proprietà	Descrizione
GameSessionArn	L'identificatore univoco della sessione di gioco della richiesta annullata. Tipo: `string` Campo obbligatorio: sì
MatchmakingConfigurationArn	L'identificatore univoco del matchmaker a cui è stata inviata questa richiesta. Tipo: `string` Campo obbligatorio: sì
TicketId	L'identificatore univoco del ticket di richiesta di riempimento da annullare. Tipo: `string` Campo obbligatorio: sì

GetFleetRoleCredentialsRequest

Questo tipo di dati offre al server di gioco un accesso limitato alle altre risorse. AWS Per ulteriori informazioni, consultare Configurare un ruolo di servizio IAM per Amazon GameLift Servers.

Proprietà	Descrizione
RoleArn	L'Amazon Resource Name (ARN) del ruolo di servizio che estende l'accesso limitato alle tue AWS risorse. Tipo: `string` Campo obbligatorio: sì
RoleSessionName	Il nome della sessione che descrive l'uso delle credenziali del ruolo. Tipo: `string` Required: No

AttributeValue

Usa questi valori nelle coppie chiave-valore degli attributi. Player Questo oggetto consente di specificare il valore di un attributo utilizzando uno qualsiasi dei tipi di dati validi: stringa, numero, array di stringhe o mappa di dati. Ogni `AttributeValue` oggetto può utilizzare solo una delle proprietà disponibili.

Proprietà	Descrizione
AttrType	Speciifica il tipo di valore dell'attributo. Tipo: un valore `AttrType` enum. Required: No
S	Rappresenta un valore di attributo di tipo stringa. Tipo: `string`

Proprietà	Descrizione
	Campo obbligatorio: sì
N	Rappresenta un valore di attributo numerico. Tipo: `double` Campo obbligatorio: sì
SL	Rappresenta una matrice di valori di attributi di tipo stringa. Tipo: `string[]` Campo obbligatorio: sì
SDM	Rappresenta un dizionario di chiavi di stringa e valori doppi. Tipo: `Dictionary<string, double>` Campo obbligatorio: sì

AwsStringOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: `string` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo.

Proprietà	Descrizione
	Tipo: the section called “GameLiftError” Required: No

GenericOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

DescribePlayerSessionsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “DescribePlayerSessionsResult” Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool`

Proprietà	Descrizione
	Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

DescribePlayerSessionsResult

Proprietà	Descrizione
NextToken	Il token che indica l'inizio della pagina successiva dei risultati. Per specificare l'inizio del set di risultati, non fornire un valore. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `string` Campo obbligatorio: sì
PlayerSessions	Una raccolta di oggetti contenente proprietà per ogni sessione di giocatore che corrisponde alla richiesta. Tipo: `IList<`the section called “PlayerSession” `>` Campo obbligatorio:
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError”

Proprietà	Descrizione
	Required: No

PlayerSession

Proprietà	Descrizione
CreationTime	Tipo: `long` Campo obbligatorio: sì
FleetId	Tipo: `string` Campo obbligatorio: sì
GameSessionId	Tipo: `string` Campo obbligatorio: sì
IpAddress	Tipo: `string` Campo obbligatorio: sì
PlayerData	Tipo: `string` Campo obbligatorio: sì
PlayerId	Tipo: `string` Campo obbligatorio: sì
PlayerSessionId	Tipo: `string` Campo obbligatorio: sì
Porta	Tipo: `int` Campo obbligatorio: sì
Stato	Tipo: Un `PlayerSessionStatus` enum.

Proprietà	Descrizione
	Campo obbligatorio: sì
TerminationTime	Tipo: `long` Campo obbligatorio: sì
DnsName	Tipo: `string` Campo obbligatorio: sì

StartMatchBackfillOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “StartMatchBackfillResult” Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

StartMatchBackfillResult

Proprietà	Descrizione
TicketId	Tipo: `string` Campo obbligatorio: sì

GetComputeCertificateOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “GetComputeCertificateResult” Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

GetComputeCertificateResult

Il percorso del certificato TLS sul computer e il nome host del computer.

Proprietà	Descrizione
CertificatePath	Tipo: `string`

Proprietà	Descrizione
	Campo obbligatorio: sì
ComputeName	Tipo: `string` Campo obbligatorio: sì

GetFleetRoleCredentialsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “GetFleetRoleCredentialsResult” Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

GetFleetRoleCredentialsResult

Proprietà	Descrizione
AccessKeyId	L'ID della chiave di accesso per autenticare e fornire l'accesso alle tue risorse. AWS Tipo: `string`

Proprietà	Descrizione
	Required: No
AssumedRoleId	L'ID dell'utente a cui appartiene il ruolo di servizio. Tipo: `string` Required: No
AssumedRoleUserArn	L'Amazon Resource Name (ARN) dell'utente a cui appartiene il ruolo di servizio. Tipo: `string` Required: No
Scadenza	Il periodo di tempo che manca alla scadenza delle credenziali di sessione. Tipo: `DateTime` Required: No
SecretAccessKey	L'ID della chiave di accesso segreta per l'autenticazione. Tipo: `string` Required: No
SessionToken	Un token per identificare la sessione attiva corrente che interagis ce con AWS le tue risorse. Tipo: `string` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì

Proprietà	Descrizione
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

AwsDateTimeOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: `DateTime` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

GameLiftError

Proprietà	Descrizione
ErrorType	Il tipo di errore. Tipo: Un enum. **`GameLiftErrorType`**

Proprietà	Descrizione
	Required: No
ErrorName	Il nome dell'errore. Tipo: `string` Required: No
ErrorMessage	Messaggio di errore. Tipo: `string` Required: No

enumerazioni;

Le enumerazioni definite per l'SDK del server per Amazon GameLift Servers (C#) sono definite come segue:

AttrType

- NONE
- STRINGA
- DOPPIA
- ELENCO_STRINGHE
- STRING_DOUBLE_MAP

GameLiftErrorType

Valore di stringa che indica il tipo di errore. I valori validi includono:

- SERVICE_CALL_FAILED — Una chiamata a un servizio non è riuscita. AWS
- LOCAL_CONNECTION_FAILED — La connessione locale a non è riuscita. Amazon GameLift Servers
- NETWORK_NOT_INITIALIZED — La rete non è stata inizializzata.
- GAMESESSION_ID_NOT_SET — L'ID della sessione di gioco non è stato impostato.
- BAD_REQUEST_EXCEPTION
- ECCEZIONE_DI_SERVIZIO INTERNA

- ALREADY_INITIALIZED — Il Amazon GameLift Servers server o il client è già stato inizializzato con Initialize ().
- FLEET_MISMATCH — La flotta bersaglio non corrisponde alla flotta di una GameSession o PlayerSession.
- GAMELIFT_CLIENT_NOT_INITIALIZED — Il client non è stato inizializzato. Amazon GameLift Servers
- GAMELIFT_SERVER_NOT_INITIALIZED — Amazon GameLift Servers Il server non è stato inizializzato.
- GAME_SESSION_ENDED_FAILED: l'SDK del server non è riuscito a contattare il servizio per segnalare la fine della sessione di gioco. Amazon GameLift Servers
- GAME_SESSION_NOT_READY — La sessione di gioco sul server non è stata attivata. Amazon GameLift Servers
- GAME_SESSION_READY_FAILED: l'SDK del server per cui non è stato possibile contattare il servizio per segnalare che la sessione di gioco è pronta. Amazon GameLift Servers
- INITIALIZATION_MISMATCH — È stato chiamato un metodo client dopo Server: :Initialize () o viceversa.
- NOT_INITIALIZED — Il Amazon GameLift Servers server o il client non sono stati inizializzati con Initialize ().
- NO_TARGET_ALIASID_SET — Non è stato impostato un aliasID di destinazione.
- NO_TARGET_FLEET_SET — Non è stata impostata una flotta di obiettivi.
- PROCESS_ENDING_FAILED: l'SDK del server non è riuscito a contattare il servizio per segnalare la fine del processo. Amazon GameLift Servers
- PROCESS_NOT_ACTIVE — Il processo del server non è ancora attivo, non è associato a un processo e non può accettarlo o elaborarlo. GameSession PlayerSessions
- PROCESS_NOT_READY — Il processo del server non è ancora pronto per essere attivato.
- PROCESS_READY_FAILED: l'SDK del server non è riuscito a contattare il servizio per Amazon GameLift Servers segnalare che il processo è pronto.
- SDK_VERSION_DETECTION_FAILED — Rilevamento della versione SDK non riuscito.
- STX_CALL_FAILED — Una chiamata al componente di backend del server non è riuscita. XStx
- STX_INITIALIZATION_FAILED — L'inizializzazione del componente di backend del server non è riuscita. XStx
- UNEXPECTED_PLAYER_SESSION — Il server ha rilevato una sessione di giocatore non registrata.

- WEBSOCKET_CONNECT_FAILURE
- WEBSOCKET_CONNECT_FAILURE_FORBIDDEN
- URL NON VALIDO DI WEBSOCKET_CONNECT_FAILURE_NON VALIDO
- WEBSOCKET_CONNECT_FAILURE_TIMEOUT
- WEBSOCKET_RETRIABLE_SEND_MESSAGE_FAILURE — Errore recuperabile nell'invio di un messaggio al servizio. GameLift WebSocket
- WEBSOCKET_SEND_MESSAGE_FAILURE — Errore nell'invio di un messaggio al GameLift servizio. WebSocket
- MATCH_BACKFILL_REQUEST_VALIDATION — La convalida della richiesta non è riuscita.
- PLAYER_SESSION_REQUEST_VALIDATION — La convalida della richiesta non è riuscita.

PlayerSessionCreationPolicy

Valore della stringa che indica se la sessione di gioco accetta nuovi giocatori. I valori validi includono:

- ACCEPT_ALL: accetta tutte le nuove sessioni giocatore.
- DENY_ALL: rifiuta tutte le nuove sessioni giocatore.
- NOT_SET — La sessione di gioco non è impostata per accettare o rifiutare sessioni di nuovi giocatori.

PlayerSessionStatus

- ACTIVE
- COMPLETED
- NOT_SET
- RISERVATO
- TIMEOUT

Server C# SDK 5.x per Amazon GameLift Servers -- Tipi di dati

Argomenti

- ProcessEnding()
- ActivateGameSession()
- UpdatePlayerSessionCreationPolicy()
- GetGameSessionId()
- GetTerminationTime()
- AcceptPlayerSession()
- RemovePlayerSession()
- DescribePlayerSessions()
- StartMatchBackfill()
- StopMatchBackfill()
- GetComputeCertificate()
- GetFleetRoleCredentials()
- Distruggi ()

GetSdkVersion()

Restituisce il numero di versione corrente dell'SDK integrato nel processo del server.

Sintassi

```
AwsStringOutcome GetSdkVersion();
```

Valore restituito

Se l'esito è positivo, restituisce la versione corrente dell'SDK come oggetto the section called "AwsStringOutcome". La stringa restituita include il numero di versione (esempio`5.0.0`). Se l'esito è negativo, verrà restituito un messaggio di errore.

Esempio

```
var getSdkVersionOutcome = GameLiftServerAPI.GetSdkVersion();
```

InitSDK()

Inizializza l'Amazon GameLift ServersSDK per una flotta gestita. EC2 Richiama questo metodo all'avvio, prima che si verifichi qualsiasi altra inizializzazione correlata a. Amazon GameLift Servers

Questo metodo legge i parametri del server dall'ambiente host per configurare la comunicazione tra il server e il Amazon GameLift Servers servizio.

Sintassi

```
GenericOutcome InitSDK();
```

Valore restituito

In caso di successo, restituisce un InitSdkOutcome oggetto per indicare che il processo del server è pronto per la chiamataProcessReady().

Esempio

```
//Call InitSDK to establish a local connection with the GameLift agent to enable
 further communication.
GenericOutcome initSDKOutcome = GameLiftServerAPI.InitSDK();
```

InitSDK()

Inizializza l'Amazon GameLift ServersSDK per una flotta Anywhere. Richiama questo metodo all'avvio, prima che si verifichi qualsiasi altra inizializzazione correlata a. Amazon GameLift Servers Questo metodo richiede parametri espliciti del server per configurare la comunicazione tra il server e il Amazon GameLift Servers servizio.

Sintassi

```
GenericOutcome InitSDK(ServerParameters serverParameters);
```

Parametri

ServerParameters

Per inizializzare un server di gioco su una flotta Amazon GameLift Servers Anywhere, costruisci un `ServerParameters` oggetto con le seguenti informazioni:

- L'URL WebSocket utilizzato per connettersi al server di gioco.
- L'ID del processo utilizzato per ospitare il server di gioco.
- L'ID del computer che ospita i processi del server di gioco.

- L'ID della Amazon GameLift Servers flotta contenente il tuo computer Amazon GameLift Servers Anywhere.
- Il token di autorizzazione generato dall'Amazon GameLift Serversoperazione.

Valore restituito

In caso di successo, restituisce un InitSdkOutcome oggetto per indicare che il processo del server è pronto per la chiamataProcessReady().

> **Note**
>
> Se le chiamate a non `InitSDK()` riescono per le build di gioco distribuite sulle flotte Anywhere, controlla il `ServerSdkVersion` parametro utilizzato durante la creazione della risorsa di compilazione. È necessario impostare esplicitamente questo valore sulla versione SDK del server in uso. Il valore predefinito per questo parametro è 4.x, che non è compatibile. Per risolvere questo problema, crea una nuova build e distribuiscila in una nuova flotta.

Esempio

```
//Define the server parameters
string websocketUrl = "wss://us-west-1.api.amazongamelift.com";
string processId = "PID1234";
string fleetId = "aarn:aws:gamelift:us-west-1:111122223333:fleet/
fleet-9999ffff-88ee-77dd-66cc-5555bbbb44aa";
string hostId = "HardwareAnywhere";
string authToken = "1111aaaa-22bb-33cc-44dd-5555eeee66ff";
ServerParameters serverParameters =
  new ServerParameters(webSocketUrl, processId, hostId, fleetId, authToken);

//Call InitSDK to establish a local connection with the GameLift agent to enable
 further communication.
GenericOutcome initSDKOutcome = GameLiftServerAPI.InitSDK(serverParameters);
```

ProcessReady()

Notifica Amazon GameLift Servers che il processo del server è pronto per ospitare sessioni di gioco. Chiama questo metodo dopo averlo InitSDK() invocato. Questo metodo deve essere chiamato solo una volta per processo.

Sintassi

```
GenericOutcome ProcessReady(ProcessParameters processParameters)
```

Parametri

ProcessParameters

Un `ProcessParameters` oggetto contiene informazioni sul processo del server.

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra sia l'implementazione del metodo che quella delle funzioni di delega.

```
// Set parameters and call ProcessReady
ProcessParameters processParams = new ProcessParameters(
  this.OnStartGameSession,
  this.OnProcessTerminate,
  this.OnHealthCheck,
  this.OnUpdateGameSession,
  port,
  new LogParameters(new List<string>()
  // Examples of log and error files written by the game server
  {
    "C:\\game\\logs",
    "C:\\game\\error"
  })
);
GenericOutcome processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);
```

ProcessEnding()

Notifica Amazon GameLift Servers che il processo del server sta terminando. Chiama questo metodo dopo tutte le altre attività di pulizia (inclusa la chiusura della sessione di gioco attiva) e prima di terminare il processo. A seconda del risultato di`ProcessEnding()`, il processo termina con successo (0) o errore (-1) e genera un evento relativo alla flotta. Se il processo termina con un errore, l'evento fleet generato è. `SERVER_PROCESS_TERMINATED_UNHEALTHY`

Sintassi

```
GenericOutcome ProcessEnding()
```

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio chiama `ProcessEnding()` e `Destroy()` prima di terminare il processo del server con un codice di uscita riuscito o di errore.

```
GenericOutcome processEndingOutcome = GameLiftServerAPI.ProcessEnding();
GameLiftServerAPI.Destroy();

if (processEndingOutcome.Success)
  {
    Environment.Exit(0);
  }
else
  {
    Console.WriteLine("ProcessEnding() failed. Error: " +
 processEndingOutcome.Error.ToString());
    Environment.Exit(-1);
  }
```

ActivateGameSession()

Notifica Amazon GameLift Servers che il processo del server ha attivato una sessione di gioco ed è ora pronto a ricevere le connessioni dei giocatori. Questa azione dovrebbe essere richiamata come parte della funzione di on`StartGameSession()` callback, dopo l'inizializzazione di tutta la sessione di gioco.

Sintassi

```
GenericOutcome ActivateGameSession()
```

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra la chiamata a `ActivateGameSession()` nell'ambito della funzione delegata on`StartGameSession()`.

```
void OnStartGameSession(GameSession gameSession)
{
  // game-specific tasks when starting a new game session, such as loading map
  // When ready to receive players
  GenericOutcome activateGameSessionOutcome = GameLiftServerAPI.ActivateGameSession();
}
```

UpdatePlayerSessionCreationPolicy()

Aggiorna la capacità della sessione di gioco corrente di accettare nuove sessioni giocatore. Una sessione di gioco può essere configurata per accettare o rifiutare tutte le nuove sessioni giocatore.

Sintassi

```
GenericOutcome UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy
 playerSessionPolicy)
```

Parametri

playerSessionPolicy

Valore di stringa che indica se la sessione di gioco accetta nuovi giocatori.

I valori validi includono:

- ACCEPT_ALL: accetta tutte le nuove sessioni giocatore.
- DENY_ALL: rifiuta tutte le nuove sessioni giocatore.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio definisce la policy di partecipazione alla sessione di gioco corrente per accettare tutti i giocatori.

```
GenericOutcome updatePlayerSessionPolicyOutcome =

 GameLiftServerAPI.UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy.ACCEPT_ALL);
```

GetGameSessionId()

Recupera l'ID della sessione di gioco ospitata dal processo del server attivo.

Per i processi inattivi che non vengono attivati con una sessione di gioco, la chiamata restituisce un. the section called “GameLiftError”

Sintassi

```
AwsStringOutcome GetGameSessionId()
```

Valore restituito

Se l'esito è positivo, l'ID della sessione di gioco verrà restituito come oggetto the section called “AwsStringOutcome”. Se non riesce, restituisce un messaggio di errore».

Esempio

```
AwsStringOutcome getGameSessionIdOutcome = GameLiftServerAPI.GetGameSessionId();
```

GetTerminationTime()

Restituisce il tempo di arresto pianificato di un processo del server, se è disponibile un tempo di chiusura. Un processo server esegue questa azione dopo aver ricevuto una `onProcessTerminate()` richiamata daAmazon GameLift Servers. Amazon GameLift Serverschiama `onProcessTerminate()` per i seguenti motivi:

- Quando il processo del server ha segnalato problemi di salute o non ha risposto. Amazon GameLift Servers
- Quando si termina l'istanza durante un evento di scale-down.
- Quando un'istanza viene terminata a causa di un'interruzione di un'istanza spot.

Sintassi

```
AwsDateTimeOutcome GetTerminationTime()
```

Valore restituito

In caso di successo, restituisce l'ora di terminazione come oggetto. the section called "AwsDateTimeOutcome" Il valore è il tempo di terminazione, espresso in segni di spunta trascorsi da allora. `0001 00:00:00` Ad esempio, il valore della data e dell'ora è uguale ai segni di `2020-09-13 12:26:40 -000Z` spunta. `637355968000000000` Se non è disponibile alcun orario di terminazione, restituisce un messaggio di errore.

Esempio

```
AwsDateTimeOutcome getTerminationTimeOutcome = GameLiftServerAPI.GetTerminationTime();
```

AcceptPlayerSession()

Notifica Amazon GameLift Servers che un giocatore con l'ID di sessione specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift Serversverifica che l'ID della sessione del giocatore sia valido. Dopo la convalida della sessione di gioco, Amazon GameLift Servers modifica lo stato dello slot del giocatore da RISERVATO ad ATTIVO.

Sintassi

```
GenericOutcome AcceptPlayerSession(String playerSessionId)
```

Parametri

playerSessionId

ID univoco rilasciato GameLift quando viene creata una nuova sessione giocatore.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra una funzione per la gestione di una richiesta di connessione, inclusa la convalida e il rifiuto di una sessione giocatore non valida. IDs

```
void ReceiveConnectingPlayerSessionID (Connection connection, String playerSessionId)
{
```

```
  GenericOutcome acceptPlayerSessionOutcome =
 GameLiftServerAPI.AcceptPlayerSession(playerSessionId);
  if(acceptPlayerSessionOutcome.Success)
  {
    connectionToSessionMap.emplace(connection, playerSessionId);
    connection.Accept();
  }
  else
  {
    connection.Reject(acceptPlayerSessionOutcome.Error.ErrorMessage);
  }
}
```

RemovePlayerSession()

Notifica Amazon GameLift Servers che un giocatore si è disconnesso dal processo del server. In risposta, Amazon GameLift Servers modifica lo slot del giocatore rendendolo disponibile.

Sintassi

```
GenericOutcome RemovePlayerSession(String playerSessionId)
```

Parametri

playerSessionId

ID univoco rilasciato Amazon GameLift Servers quando viene creata una nuova sessione di gioco.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
GenericOutcome removePlayerSessionOutcome =
 GameLiftServerAPI.RemovePlayerSession(playerSessionId);
```

DescribePlayerSessions()

Recupera i dati della sessione del giocatore che includono impostazioni, metadati della sessione e dati del giocatore. Utilizza questa operazione per ottenere le informazioni per una singola sessione

giocatore, per tutte le sessioni giocatore in una sessione di gioco o per tutte le sessioni giocatore associate a un singolo ID giocatore.

Sintassi

```
DescribePlayerSessionsOutcome DescribePlayerSessions(DescribePlayerSessionsRequest
 describePlayerSessionsRequest)
```

Parametri

DescribePlayerSessionsRequest

Un the section called “DescribePlayerSessionsRequest” oggetto che descrive le sessioni dei giocatori da recuperare.

Valore restituito

In caso di successo, restituisce un the section called “DescribePlayerSessionsOutcome” oggetto che contiene un set di oggetti della sessione del giocatore che soddisfano i parametri della richiesta.

Esempio

Questo esempio illustra una richiesta per tutte le sessioni giocatore attivamente connesse a una sessione di gioco specificata. Omettendo NextTokene impostando il valore Limite su 10, Amazon GameLift Servers restituirà i primi 10 record della sessione di gioco corrispondenti alla richiesta.

```
// Set request parameters
DescribePlayerSessionsRequest describePlayerSessionsRequest = new
 DescribePlayerSessionsRequest()
{
  GameSessionId = GameLiftServerAPI.GetGameSessionId().Result,    //gets the ID for the
 current game session
  Limit = 10,
  PlayerSessionStatusFilter =
    PlayerSessionStatusMapper.GetNameForPlayerSessionStatus(PlayerSessionStatus.ACTIVE)
};
// Call DescribePlayerSessions
DescribePlayerSessionsOutcome describePlayerSessionsOutcome =
  GameLiftServerAPI.DescribePlayerSessions(describePlayerSessionsRequest);
```

StartMatchBackfill()

Invia una richiesta per trovare nuovi giocatori per gli slot aperti in una sessione di gioco creata con FlexMatch. Per ulteriori informazioni, consulta la funzione di FlexMatchriempimento.

Questa operazione è asincrona. Se vengono abbinati nuovi giocatori, Amazon GameLift Servers fornisce dati aggiornati sul matchmaker utilizzando la funzione di callback. `OnUpdateGameSession()`

Un processo del server può avere un solo backfill degli abbinamenti attivo alla volta. Per inviare una nuova richiesta, chiama prima StopMatchBackfill() per annullare la richiesta originale.

Sintassi

```
StartMatchBackfillOutcome StartMatchBackfill (StartMatchBackfillRequest
 startBackfillRequest);
```

Parametri

StartMatchBackfillRequest

Un `StartMatchBackfillRequest` oggetto contiene informazioni sulla richiesta di riempimento.

Valore restituito

Restituisce un the section called “StartMatchBackfillOutcome” oggetto con l'ID del ticket Match Backfill o restituisce un errore con un messaggio di errore.

Esempio

```
// Build a backfill request
StartMatchBackfillRequest startBackfillRequest = new StartMatchBackfillRequest()
{
  TicketId = "1111aaaa-22bb-33cc-44dd-5555eeee66ff", //optional
  MatchmakingConfigurationArn = "arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig",
  GameSessionId = GameLiftServerAPI.GetGameSessionId().Result,    // gets ID for
 current game session
  MatchmakerData matchmakerData =
    MatchmakerData.FromJson(gameSession.MatchmakerData),  // gets matchmaker data for
 current players
```

```
  // get matchmakerData.Players
  // remove data for players who are no longer connected
  Players = ListOfPlayersRemainingInTheGame
};

// Send backfill request
StartMatchBackfillOutcome startBackfillOutcome =
 GameLiftServerAPI.StartMatchBackfill(startBackfillRequest);

// Implement callback function for backfill
void OnUpdateGameSession(GameSession myGameSession)
{
  // game-specific tasks to prepare for the newly matched players and update matchmaker
 data as needed
}
```

StopMatchBackfill()

Annulla una richiesta di match backfill attiva. Per ulteriori informazioni, consulta la funzione di FlexMatchriempimento.

Sintassi

```
GenericOutcome StopMatchBackfill (StopMatchBackfillRequest stopBackfillRequest);
```

Parametri

StopMatchBackfillRequest

Un `StopMatchBackfillRequest` oggetto che fornisce dettagli sul ticket di matchmaking che stai interrompendo.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
// Set backfill stop request parameters
StopMatchBackfillRequest stopBackfillRequest = new StopMatchBackfillRequest(){
  TicketId = "1111aaaa-22bb-33cc-44dd-5555eeee66ff", //optional, if not provided one is
 autogenerated
```

```
  MatchmakingConfigurationArn = "arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig",
  GameSessionId = GameLiftServerAPI.GetGameSessionId().Result    //gets the ID for the
 current game session
};
GenericOutcome stopBackfillOutcome =
 GameLiftServerAPI.StopMatchBackfillRequest(stopBackfillRequest);
```

GetComputeCertificate()

Recupera il percorso del certificato TLS utilizzato per crittografare la connessione di rete tra il server di gioco e il client di gioco. Puoi utilizzare il percorso del certificato quando registri il tuo dispositivo di elaborazione in una flotta Anywhere. Amazon GameLift Servers Per ulteriori informazioni, consultare RegisterCompute.

Sintassi

```
GetComputeCertificateOutcome GetComputeCertificate();
```

Valore restituito

Restituisce un GetComputeCertificateResponse oggetto che contiene quanto segue:

- CertificatePath: il percorso del certificato TLS sulla risorsa di calcolo. Quando si utilizza una flotta Amazon GameLift Servers gestita, questo percorso contiene:
  - `certificate.pem`: Il certificato dell'utente finale. La catena completa di certificati è la combinazione dei certificati `certificateChain.pem` aggiunti a questo certificato.
  - `certificateChain.pem`: La catena di certificati che contiene il certificato principale e i certificati intermedi.
  - `rootCertificate.pem`: Il certificato principale.
  - `privateKey.pem`: la chiave privata per il certificato dell'utente finale.
- ComputeName: il nome della risorsa di calcolo.

Esempio

```
GetComputeCertificateOutcome getComputeCertificateOutcome =
 GameLiftServerAPI.GetComputeCertificate();
```

GetFleetRoleCredentials()

Recupera le credenziali del ruolo IAM che autorizzano Amazon GameLift Servers a interagire con altri. Servizi AWS Per ulteriori informazioni, consulta Comunica con altre AWS risorse delle tue flotte.

Sintassi

```
GetFleetRoleCredentialsOutcome GetFleetRoleCredentials(GetFleetRoleCredentialsRequest
 request);
```

Parametri

GetFleetRoleCredentialsRequest

Credenziali di ruolo che estendono l'accesso limitato alle tue AWS risorse al server di gioco.

Valore restituito

Restituisce un oggetto the section called “GetFleetRoleCredentialsOutcome”.

Esempio

```
// form the fleet credentials request
GetFleetRoleCredentialsRequest getFleetRoleCredentialsRequest = new
 GetFleetRoleCredentialsRequest(){
  RoleArn = "arn:aws:iam::123456789012:role/service-role/exampleGameLiftAction"
};
GetFleetRoleCredentialsOutcome GetFleetRoleCredentialsOutcome credentials =
 GetFleetRoleCredentials(getFleetRoleCredentialsRequest);
```

Distruggi ()

Libera l'SDK del server di Amazon GameLift Servers gioco dalla memoria. È consigliabile chiamare questo metodo dopo `ProcessEnding()` e prima di terminare il processo. Se utilizzi una flotta Anywhere e non interrompi i processi del server dopo ogni sessione di gioco, chiama `Destroy()` e poi `InitSDK()` reinizializza prima di notificare Amazon GameLift Servers che il processo è pronto per ospitare una sessione di gioco. `ProcessReady()`

Sintassi

```
GenericOutcome Destroy()
```

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
// Operations to end game sessions and the server process
GenericOutcome processEndingOutcome = GameLiftServerAPI.ProcessEnding();

// Shut down and destroy the instance of the GameLift Game Server SDK
GenericOutcome destroyOutcome = GameLiftServerAPI.Destroy();

// Exit the process with success or failure
if (processEndingOutcome.Success)
  {
    Environment.Exit(0);
  }
else
  {
    Console.WriteLine("ProcessEnding() failed. Error: " +
 processEndingOutcome.Error.ToString());
    Environment.Exit(-1);
  }
```

Go server SDK per Amazon GameLift Servers -- Azioni

Usa il riferimento all'SDK 5.x del server per integrare il tuo gioco multiplayer con cui ospitarlo. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consulta. Aggiungi Amazon GameLift Servers al tuo server di gioco

`GameLiftServerAPI.go`definisce le azioni dell'SDK del server Go.

Go server SDK perAmazon GameLift Servers: tipi di dati

Usa il riferimento all'SDK del server per integrare il tuo gioco multiplayer con cui ospitarlo. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

`GameLiftServerAPI.go`definisce le azioni dell'SDK del server Go.

Go server SDK per Amazon GameLift Servers -- Azioni

Tipi di dati

- LogParameters
- ProcessParameters
- UpdateGameSession
- GameSession
- ServerParameters
- StartMatchBackfillRequest
- Player
- DescribePlayerSessionsRequest
- StopMatchBackfillRequest
- GetFleetRoleCredentialsRequest

LogParameters

Un oggetto che identifica i file generati durante una sessione di gioco che desideri caricare e archiviare Amazon GameLift Servers al termine della sessione di gioco. Il server di gioco `LogParameters` lo fornisce Amazon GameLift Servers come parte di un `ProcessParameters` oggetto in una ProcessReady() chiamata.

Proprietà	Descrizione
`LogPaths`	L'elenco dei percorsi di directory dei file di registro del server di gioco che desideri Amazon GameLift Servers archiviare per accessi futuri. Il processo del server genera questi file durante ogni sessione di gioco. Definisci i percorsi e i nomi dei file nel tuo server di gioco e li memorizzi nella directory di build del gioco principale. I percorsi di registro devono essere assoluti. Ad esempio, se la build del gioco memorizza i log delle sessioni di gioco in un percorso come`MyGame\sessionLogs\` , il percorso si troverebbe `c:\game\MyGame\sessionLogs` su un'istanza di Windows. Tipo: `[]string` Required: No

ProcessParameters

Un oggetto che descrive la comunicazione tra un processo server e. Amazon GameLift Servers Il processo server fornisce queste informazioni a Amazon GameLift Servers con una chiamata aProcessReady().

Proprietà	Descrizione
`LogParameters`	Un oggetto con percorsi di directory ai file generati durante una sessione di gioco. Amazon GameLift Serverscopia e archivia i file per accessi futuri. Tipo: `LogParameters` Required: No
`OnHealthCheck`	La funzione di callback che Amazon GameLift Servers richiama per richieder e un rapporto sullo stato di salute del processo del server. Amazon GameLift Serverschiama questa funzione ogni 60 secondi e attende una risposta per 60 secondi. Il processo del server ritorna `TRUE` se integro, `FALSE` se non integro. Se non viene restituita alcuna risposta, Amazon GameLift Servers registra il processo del server come non integro. Tipo: `OnHealthCheck func() bool` Required: No
`OnProcessTerminate`	La funzione di callback che Amazon GameLift Servers richiama per forzare l'arresto del processo server. Dopo aver chiamato questa funzione, Amazon GameLift Servers attende 5 minuti che il processo server si spenga e risponda con una ProcessEnding() chiamata prima di chiudere il processo server. Tipo: `OnProcessTerminate func()` Campo obbligatorio: sì
`OnStartGameSession`	La funzione di callback che Amazon GameLift Servers richiama il passaggio di un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift Serverschiama questa funzione quando è stata elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un

	GameSession oggetto, uno status update (`updateReason`) e l'ID del ticket match backfill. Tipo: `OnStartGameSession func (model.GameSession )` Campo obbligatorio: sì
`OnUpdateGameSession`	La funzione di callback che Amazon GameLift Servers richiama per passare informazioni aggiornate sulla sessione di gioco al processo del server. Amazon GameLift Serverschiama questa funzione dopo aver elaborato una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Tipo: `OnUpdateGameSession func (model.UpdateGameSession)` Required: No
`Port`	Il numero di porta su cui il processo del server ascolta le connessioni di nuovi giocatori. Il valore deve rientrare nella gamma di porte configurate per qualsiasi parco istanze che distribuisce questa build del server di gioco. Questo numero di porta è incluso nella sessione di gioco e negli oggetti della sessione del giocatore utilizzati dalle sessioni di gioco per la connessione a un processo server. Tipo: `int` Campo obbligatorio: sì

UpdateGameSession

Gli aggiornamenti a un oggetto della sessione di gioco, che include il motivo per cui la sessione di gioco è stata aggiornata e il relativo ID del ticket di backfill se il backfill viene utilizzato per riempire le sessioni dei giocatori nella sessione di gioco.

Proprietà	Descrizione
GameSession	Oggetto GameSession. L'`GameSession` oggetto contiene proprietà che descrivono una sessione di gioco.

Proprietà	Descrizione
	Tipo: `GameSession GameSession()` Campo obbligatorio: sì
UpdateReason	Il motivo per cui la sessione di gioco viene aggiornata. Tipo: `UpdateReason UpdateReason()` Campo obbligatorio: sì
BackfillTicketId	L'ID del ticket di backfill che tenta di aggiornare la sessione di gioco. Tipo: `String` Required: No

GameSession

I dettagli di una sessione di gioco.

Proprietà	Descrizione
GameSessionId	Un identificatore univoco per la sessione di gioco. Una sessione di gioco Amazon Resource Name (ARN) ha il seguente formato:. `arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token>` Tipo: `String` Required: No
Nome	Un'etichetta descrittiva della sessione di gioco. Tipo: `String` Required: No
FleetId	Un identificatore univoco per la flotta su cui è in esecuzione la sessione di gioco.

Proprietà	Descrizione
	Tipo: `String` Required: No
MaximumPlayerSessionCount	Il numero massimo di connessioni dei giocatori alla sessione di gioco. Tipo: `Integer` Required: No
Porta	Il numero di porta per la sessione di gioco. Per connettersi a un server di Amazon GameLift Servers gioco, un'app necessita sia dell'indirizzo IP che del numero di porta. Tipo: `Integer` Required: No
IpAddress	L'indirizzo IP della sessione di gioco. Per connettersi a un server di Amazon GameLift Servers gioco, un'app necessita sia dell'indirizzo IP che del numero di porta. Tipo: `String` Required: No
GameSessionData	Un set di proprietà personalizzate della sessione di gioco, formattate come valore di stringa singola. Tipo: `String` Required: No

Proprietà	Descrizione
MatchmakerData	Le informazioni sul processo di matchmaking utilizzato per creare la sessione di gioco, in sintassi JSON, sono formattate come stringa. Oltre alla configura zione di matchmaking utilizzata, contiene dati su tutti i giocatori assegnati alla partita, compresi gli attributi dei giocatori e le assegnazioni delle squadre. Tipo: `String` Required: No
GameProperties	Un insieme di proprietà personalizzate per una sessione di gioco, formattat e come coppie chiave:valore. Queste proprietà vengono passate con una richiesta di avvio di una nuova sessione di gioco. Tipo: `map[string] string` Required: No
DnsName	L'identificatore DNS assegnato all'istanza che esegue la sessione di gioco. I valori hanno il seguente formato: • Flotte abilitate per TLS:. `<unique identifier>.<region identifier>.amazongamelift.com` • Non-TLS-enabled flotte:. `ec2-<unique identifier>.compute.amazonaws.com` Quando ti connetti a una sessione di gioco in esecuzione su una flotta abilitata per TLS, devi usare il nome DNS, non l'indirizzo IP. Tipo: `String` Required: No

ServerParameters

Informazioni utilizzate per mantenere la connessione tra un server Amazon GameLift Servers Anywhere e il servizio. Amazon GameLift Servers Queste informazioni vengono utilizzate quando si

avviano nuovi processi server conInitSDK(). Per i server ospitati su EC2 istanze Amazon GameLift Servers gestite, utilizza un oggetto vuoto.

Proprietà	Descrizione
`WebSocket URL`	`GameLiftServerSdkEndpoint` Amazon GameLift ServersRestituisce quando si utilizza una `RegisterCompute` risorsa di calcolo Amazon GameLift Servers Anywhere. Tipo: `string` Campo obbligatorio: sì
`ProcessID`	Un identificatore univoco registrato nel processo del server che ospita il gioco. Tipo: `string` Campo obbligatorio: sì
`HostID`	L'identificatore univoco della risorsa di elaborazione che ospita il nuovo processo del server. `HostID`Viene `ComputeName` utilizzato al momento della registrazione del computer. Per ulteriori informazioni, consulta RegisterCompute. Tipo: `string` Campo obbligatorio: sì
`FleetID`	L'identificatore univoco della flotta su cui è registrato il calcolo. Per ulteriori informazioni, consulta RegisterCompute. Tipo: `string` Campo obbligatorio: sì
`AuthToken`	Il token di autenticazione generato da Amazon GameLift Servers questo codice autentica il server. Amazon GameLift Servers Per ulteriori informazioni, consulta GetComputeAuthToken. Tipo: `string`

Proprietà	Descrizione
	Campo obbligatorio: sì

StartMatchBackfillRequest

Informazioni utilizzate per creare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni durante una chiamataAmazon GameLift Servers. StartMatchBackfill()

Proprietà	Descrizione
GameSessionArn	L'identificatore univoco della sessione di gioco. L'operazione API `GetGameSessionId` restituisce l'identificatore in formato ARN. Tipo: `String` Campo obbligatorio: sì
MatchmakingConfigurationArn	L'identificatore univoco (sotto forma di ARN) che il matchmaker deve utilizzare per questa richiesta. L'ARN del matchmaker per la sessione di gioco originale si trova nell'oggetto della sessione di gioco nella proprietà matchmaker data. Per ulteriori informazioni sui dati del matchmaker, vedi Lavora con i dati del matchmaker. Tipo: `String` Campo obbligatorio: sì
Players	Un insieme di dati che rappresenta tutti i giocatori attualmente impegnati nella sessione di gioco. Il matchmaker utilizza queste informazioni per cercare nuovi giocatori che rappresentano un buon abbinamento per i giocatori attuali. Tipo: `[]model.Player` Campo obbligatorio: sì
TicketId	L'identificatore univoco di un ticket di richiesta di matchmaking o match backfill. Se non fornisci un valore, Amazon GameLift Servers ne genera uno. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill degli abbinamenti o annullare la richiesta, se necessario.

Proprietà	Descrizione
	Tipo: `String` Required: No

Player

L'oggetto che rappresenta un giocatore in matchmaking. Quando inizia una richiesta di matchmaking, un giocatore ha un ID giocatore, attributi e possibilmente dati sulla latenza. Amazon GameLift Serversaggiunge le informazioni sulla squadra dopo una partita.

Proprietà	Descrizione
LatencyInSM	Un insieme di valori espressi in millisecondi che indica la quantità di latenza che un giocatore sperimenta quando è connesso a una posizione. Se viene utilizzata questa proprietà, il giocatore viene abbinato solo per le posizioni elencate. Se un matchmaker ha una regola che valuta la latenza dei giocatori, i giocatori devono segnalare la latenza per essere abbinati. Tipo: `map[string] int` Required: No
PlayerAttributes	Una raccolta di coppie chiave:valore che contengono informazioni sui giocatori da utilizzare nel matchmaking. Le chiavi degli attributi del giocatore devono corrispondere a quelle PlayerAttributes utilizzate in un set di regole di matchmaking. Per ulteriori informazioni sugli attributi dei giocatori, vedi AttributeValue. Tipo: `map[string] AttributeValue` Required: No
PlayerId	Un identificatore univoco per un giocatore. Tipo: `String`

Proprietà	Descrizione
	Required: No
Team	Il nome della squadra a cui è assegnato il giocatore in una partita. Definisci il nome della squadra nel set di regole del matchmaking. Tipo: `String` Required: No

DescribePlayerSessionsRequest

Un oggetto che specifica le sessioni dei giocatori da recuperare. Il processo server fornisce queste informazioni con una DescribePlayerSessions() chiamata a. Amazon GameLift Servers

Proprietà	Descrizione
`GameSessionID`	Un identificatore univoco della sessione di gioco. Utilizzare questo parametro per richiedere tutte le sessioni giocatore per la sessione di gioco specificata. Il formato dell'ID della sessione di gioco è`arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>` . `GameSessionID` È una stringa ID personalizzata o una stringa generata. Tipo: `String` Required: No
`PlayerSessionID`	L'identificatore univoco per la sessione di un giocatore. Usa questo parametro per richiedere una singola sessione di gioco specifica. Tipo: `String` Required: No
`PlayerID`	L'identificatore univoco di un giocatore. Usa questo parametro per richieder e tutte le sessioni di gioco per un giocatore specifico. Consultare Genera giocatore IDs.

Proprietà	Descrizione
	Tipo: `String` Required: No
`PlayerSessionStatusFilter`	Lo stato della sessione del giocatore su cui filtrare i risultati. I possibili stati delle sessioni di gioco includono: • RISERVATO — La richiesta di sessione del giocatore è stata ricevuta, ma il giocatore non si è connesso al processo del server né è stato convalidato. • ATTIVO: il giocatore è stato convalidato dal processo del server ed è connesso. • COMPLETATA — La connessione del giocatore è stata interrotta. • TIMEDOUT: è stata ricevuta una richiesta di sessione, ma il giocatore non si è connesso o non è stato convalidato entro il limite di timeout (60 secondi). Tipo: `String` Required: No
`NextToken`	Il token che indica l'inizio della pagina successiva di risultati. Per specificare l'inizio del set di risultati, non fornire un valore. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `String` Required: No
`Limit`	Numero massimo di risultati da restituire. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `int` Required: No

StopMatchBackfillRequest

Informazioni utilizzate per annullare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni al Amazon GameLift Servers servizio durante una chiamata. StopMatchBackfill()

Proprietà	Descrizione
GameSessionArn	L'identificatore univoco della sessione di gioco della richiesta annullata. Tipo: `string` Required: No
MatchmakingConfigurationArn	L'identificatore univoco del matchmaker a cui è stata inviata questa richiesta. Tipo: `string` Required: No
TicketId	L'identificatore univoco del ticket di richiesta di riempimento da annullare. Tipo: `string` Required: No

GetFleetRoleCredentialsRequest

Le credenziali del ruolo che estendono l'accesso limitato alle tue AWS risorse al server di gioco. Per ulteriori informazioni, consultare Configurare un ruolo di servizio IAM per Amazon GameLift Servers.

Proprietà	Descrizione
RoleArn	L'ARN del ruolo di servizio che estende l'accesso limitato alle AWS risorse. Tipo: `string` Campo obbligatorio: sì
RoleSessionName	Il nome della sessione che descrive l'uso delle credenziali del ruolo.

Proprietà	Descrizione
	Tipo: `string` Campo obbligatorio: sì

[Go server SDK perAmazon GameLift Servers: tipi di dati]

Azioni

- GetSdkVersion()
- InitSDK()
- ProcessReady()
- ProcessEnding()
- ActivateGameSession()
- UpdatePlayerSessionCreationPolicy()
- GetGameSessionId()
- GetTerminationTime()
- AcceptPlayerSession()
- RemovePlayerSession()
- DescribePlayerSessions()
- StartMatchBackfill()
- StopMatchBackfill()
- GetComputeCertificate()
- GetFleetRoleCredentials()
- Distruggi ()

GetSdkVersion()

Restituisce il numero di versione corrente dell'SDK integrato nel processo del server.

Sintassi

```
func GetSdkVersion() (string, error)
```

Valore restituito

In caso di successo, restituisce la versione SDK corrente come stringa. La stringa restituita include il numero di versione (esempio`5.0.0`). Se non riesce, restituisce un messaggio di errore come`common.SdkVersionDetectionFailed`.

Esempio

```
version, err := server.GetSdkVersion()
```

InitSDK()

Inizializza l'SDK Amazon GameLift Servers. Chiama questo metodo all'avvio prima che Amazon GameLift Servers si verifichi qualsiasi altra inizializzazione correlata a. Questo metodo imposta la comunicazione tra il server e il Amazon GameLift Servers servizio.

Sintassi

```
func InitSDK(params ServerParameters) error
```

Parametri

ServerParameters

Per inizializzare un server di gioco su una flotta Amazon GameLift Servers Anywhere, costruisci un `ServerParameters` oggetto con le seguenti informazioni:

- L'URL WebSocket utilizzato per connettersi al server di gioco.
- L'ID del processo utilizzato per ospitare il server di gioco.
- L'ID del computer che ospita i processi del server di gioco.
- L'ID della Amazon GameLift Servers flotta contenente il tuo computer Amazon GameLift Servers Anywhere.
- Il token di autorizzazione generato dall'Amazon GameLift Serversoperazione.

Per inizializzare un server di gioco su una EC2 flotta Amazon GameLift Servers gestita, costruisci un `ServerParameters` oggetto senza parametri. Con questa chiamata, l'Amazon GameLift Serversagente configura l'ambiente di calcolo e si connette automaticamente al Amazon GameLift Servers servizio per te.

Valore restituito

In caso di successo, restituisce un `nil` errore per indicare che il processo del server è pronto per la chiamataProcessReady().

> **Note**
>
> Se le chiamate a non `InitSDK()` riescono per le build di gioco distribuite sulle flotte Anywhere, controlla il `ServerSdkVersion` parametro utilizzato durante la creazione della risorsa di compilazione. È necessario impostare esplicitamente questo valore sulla versione SDK del server in uso. Il valore predefinito per questo parametro è 4.x, che non è compatibile. Per risolvere questo problema, crea una nuova build e distribuiscila in una nuova flotta.

Esempio

Amazon GameLift ServersEsempio: Ovunque

```
//Define the server parameters
serverParameters := ServerParameters {
  WebSocketURL: "wss://us-west-1.api.amazongamelift.com",
  ProcessID: "PID1234",
  HostID: "HardwareAnywhere",
  FleetID: "aarn:aws:gamelift:us-west-1:111122223333:fleet/
fleet-9999ffff-88ee-77dd-66cc-5555bbbb44aa",
  AuthToken: "1111aaaa-22bb-33cc-44dd-5555eeee66ff"
}

//Call InitSDK to establish a local connection with the Amazon GameLift Servers Agent
 to enable further communication.
err := server.InitSDK(serverParameters)
```

Amazon GameLift Servers EC2 esempio gestito

```
//Define the server parameters
serverParameters := ServerParameters {}

//Call InitSDK to establish a local connection with the Amazon GameLift Servers Agent
 to enable further communication.
err := server.InitSDK(serverParameters)
```

ProcessReady()

Notifica Amazon GameLift Servers che il processo del server è pronto per ospitare sessioni di gioco. Chiama questo metodo dopo averlo InitSDK() invocato. Questo metodo deve essere chiamato solo una volta per processo.

Sintassi

```
func ProcessReady(param ProcessParameters) error
```

Parametri

ProcessParameters

Un ProcessParameters oggetto comunica le seguenti informazioni sul processo del server:

- I nomi dei metodi di callback implementati nel codice del server di gioco richiamato dal Amazon GameLift Servers servizio per comunicare con il processo del server.
- Il numero di porta su cui il processo server è in ascolto.
- Il tipo di LogParameters dati contenente il percorso di tutti i file specifici della sessione di gioco che desideri acquisire e Amazon GameLift Servers archiviare.

Valore restituito

Restituisce un errore con un messaggio di errore se il metodo fallisce. Restituisce `nil` se il metodo ha successo.

Esempio

Questo esempio illustra la chiamata ProcessReady() e le implementazioni della funzione delegata.

```
// Define the process parameters
processParams := ProcessParameters {
  OnStartGameSession: gameProcess.OnStartGameSession,
  OnUpdateGameSession: gameProcess.OnGameSessionUpdate,
  OnProcessTerminate: gameProcess.OnProcessTerminate,
  OnHealthCheck: gameProcess.OnHealthCheck,
  Port: port,
  LogParameters: LogParameters {    // logging and error example
    []string {"C:\\game\\logs", "C:\\game\\error"}
```

```
  }
}

err := server.ProcessReady(processParams)
```

ProcessEnding()

Notifica Amazon GameLift Servers che il processo del server sta terminando. Chiama questo metodo dopo tutte le altre attività di pulizia (inclusa la chiusura della sessione di gioco attiva) e prima di terminare il processo. A seconda del risultato di`ProcessEnding()`, il processo termina con successo (0) o errore (-1) e genera un evento relativo alla flotta. Se il processo termina con un errore, l'evento fleet generato è. `SERVER_PROCESS_TERMINATED_UNHEALTHY`

Sintassi

```
func ProcessEnding() error
```

Valore restituito

Restituisce un codice di errore 0 o un codice di errore definito.

Esempio

```
// operations to end game sessions and the server process
defer func() {
  err := server.ProcessEnding()
  server.Destroy()
  if err != nil {
    fmt.Println("ProcessEnding() failed. Error: ", err)
    os.Exit(-1)
  } else {
    os.Exit(0)
  }
}
```

ActivateGameSession()

Notifica Amazon GameLift Servers che il processo del server ha attivato una sessione di gioco ed è ora pronto per ricevere le connessioni dei giocatori. Questa azione viene richiamata come parte della funzione di `onStartGameSession()` callback, dopo l'inizializzazione di tutta la sessione di gioco.

Sintassi

```
func ActivateGameSession() error
```

Valore restituito

Restituisce un errore con un messaggio di errore se il metodo fallisce.

Esempio

Questo esempio mostra le `ActivateGameSession()` chiamate come parte della funzione `onStartGameSession()` delegata.

```
func OnStartGameSession(GameSession gameSession) {
  // game-specific tasks when starting a new game session, such as loading map
  // Activate when ready to receive players
  err := server.ActivateGameSession();
}
```

UpdatePlayerSessionCreationPolicy()

Aggiorna la capacità della sessione di gioco corrente di accettare nuove sessioni giocatore. Una sessione di gioco può essere configurata per accettare o rifiutare tutte le nuove sessioni giocatore.

Sintassi

```
func UpdatePlayerSessionCreationPolicy(policy model.PlayerSessionCreationPolicy) error
```

Parametri

playerSessionCreationPolitica

Valore di stringa che indica se la sessione di gioco accetta nuovi giocatori.

I valori validi includono:

- **`model.AcceptAll`**— Accetta tutte le sessioni di nuovi giocatori.
- **`model.DenyAll`**— Nega tutte le sessioni dei nuovi giocatori.

Valore restituito

Restituisce un errore con un messaggio di errore se si verifica un errore.

Esempio

Questo esempio definisce la policy di partecipazione alla sessione di gioco corrente per accettare tutti i giocatori.

```
err := server.UpdatePlayerSessionCreationPolicy(model.AcceptAll)
```

GetGameSessionId()

Recupera l'ID della sessione di gioco ospitata dal processo del server attivo.

Sintassi

```
func GetGameSessionID() (string, error)
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce l'ID della sessione di gioco e l'errore zero. Per i processi inattivi che non sono ancora stati attivati con una sessione di gioco, la chiamata restituisce una stringa vuota e `nil` un errore.

Esempio

```
gameSessionID, err := server.GetGameSessionID()
```

GetTerminationTime()

Restituisce l'ora in cui è pianificata la chiusura di un processo del server, se è disponibile un orario di terminazione. Un processo server esegue questa azione dopo aver ricevuto una `onProcessTerminate()` richiamata da. Amazon GameLift Servers Amazon GameLift Serverschiama `onProcessTerminate()` per i seguenti motivi:

- Quando il processo del server ha segnalato problemi di salute o non ha risposto. Amazon GameLift Servers
- Quando si termina l'istanza durante un evento di scale-down.
- Quando un'istanza viene terminata a causa di un'interruzione di un'istanza spot.

Sintassi

```
func GetTerminationTime() (int64, error)
```

Valore restituito

In caso di successo, restituisce il timestamp in secondi di epoca in cui è pianificata la chiusura del processo del server e l'interruzione di un errore. `nil` Il valore è il tempo di terminazione, espresso in tick trascorsi da. `0001 00:00:00` Ad esempio, il valore di data e ora è uguale ai segni di `2020-09-13 12:26:40 -000Z` spunta. `637355968000000000` Se non è disponibile alcun orario di terminazione, restituisce un messaggio di errore.

Esempio

```
terminationTime, err := server.GetTerminationTime()
```

AcceptPlayerSession()

Notifica Amazon GameLift Servers che un giocatore con l'ID di sessione specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift Serversverifica che l'ID della sessione del giocatore sia valido. Dopo la convalida della sessione del giocatore, Amazon GameLift Servers modifica lo stato dello slot del giocatore da a`RESERVED`. `ACTIVE`

Sintassi

```
func AcceptPlayerSession(playerSessionID string) error
```

Parametri

`playerSessionId`

ID univoco rilasciato da Amazon GameLift Servers quando viene creata una nuova sessione giocatore.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio gestisce una richiesta di connessione che include la convalida e il rifiuto di una sessione giocatore non valida. IDs

```
func ReceiveConnectingPlayerSessionID(conn Connection, playerSessionID string) {
    err := server.AcceptPlayerSession(playerSessionID)
    if err != nil {
        connection.Accept()
    } else {
        connection.Reject(err.Error())
    }
}
```

RemovePlayerSession()

Notifica Amazon GameLift Servers che un giocatore si è disconnesso dal processo del server. In risposta, Amazon GameLift Servers modifica lo slot del giocatore rendendolo disponibile.

Sintassi

```
func RemovePlayerSession(playerSessionID string) error
```

Parametri

`playerSessionId`

ID univoco rilasciato Amazon GameLift Servers quando viene creata una nuova sessione giocatore.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
err := server.RemovePlayerSession(playerSessionID)
```

DescribePlayerSessions()

Recupera i dati della sessione del giocatore che includono impostazioni, metadati della sessione e dati del giocatore. Utilizzate questo metodo per ottenere informazioni su quanto segue:

- Una sessione per giocatore singolo
- Tutte le sessioni dei giocatori in una sessione di gioco
- Tutte le sessioni dei giocatori sono associate a un ID giocatore singolo

Sintassi

```
func DescribePlayerSessions(req request.DescribePlayerSessionsRequest)
 (result.DescribePlayerSessionsResult, error) {
 return srv.describePlayerSessions(&req)
}
```

Parametri

DescribePlayerSessionsRequest

Un `DescribePlayerSessionsRequest` oggetto descrive le sessioni dei giocatori da recuperare.

Valore restituito

In caso di successo, restituisce un `DescribePlayerSessionsResult` oggetto che contiene un set di oggetti della sessione del giocatore che soddisfano i parametri della richiesta.

Esempio

Questo esempio richiede che tutte le sessioni dei giocatori siano connesse attivamente a una sessione di gioco specificata. Omettendo NextTokene impostando il valore Limite su 10, Amazon GameLift Servers restituisce i primi 10 record di sessione tra giocatori corrispondenti alla richiesta.

```
// create request
describePlayerSessionsRequest := request.NewDescribePlayerSessions()
describePlayerSessionsRequest.GameSessionID, _ = server.GetGameSessionID() // get ID
 for the current game session
describePlayerSessionsRequest.Limit = 10                                  // return the
 first 10 player sessions
```

```
describePlayerSessionsRequest.PlayerSessionStatusFilter = "ACTIVE"         // Get all
 player sessions actively connected to the game session

describePlayerSessionsResult, err :=
 server.DescribePlayerSessions(describePlayerSessionsRequest)
```

StartMatchBackfill()

Invia una richiesta per trovare nuovi giocatori per gli slot aperti in una sessione di gioco creata con FlexMatch. Per ulteriori informazioni, consulta la funzione di FlexMatchriempimento.

Questa operazione è asincrona. Se vengono abbinati nuovi giocatori, Amazon GameLift Servers fornisce dati aggiornati sul matchmaker utilizzando la funzione di callback. `OnUpdateGameSession()`

Un processo del server può avere un solo backfill degli abbinamenti attivo alla volta. Per inviare una nuova richiesta, chiama prima StopMatchBackfill() per annullare la richiesta originale.

Sintassi

```
func StartMatchBackfill(req request.StartMatchBackfillRequest)
 (result.StartMatchBackfillResult, error)
```

Parametri

StartMatchBackfillRequest

Un StartMatchBackfillRequest oggetto comunica le seguenti informazioni:

- ID del ticket da assegnare alla richiesta di backfill. Queste informazioni sono facoltative; se non viene fornito alcun ID, ne Amazon GameLift Servers genera uno.
- Matchmaker a cui inviare la richiesta. L'ARN di configurazione completo è obbligatorio. Questo valore si trova nei dati del matchmaker della sessione di gioco.
- L'ID della sessione di gioco da riempire.
- I dati di matchmaking disponibili per i giocatori attuali della sessione di gioco.

Valore restituito

Restituisce un `StartMatchBackfillResult` oggetto con l'ID del ticket match backfill, oppure restituisce un errore con un messaggio di errore.

Esempio

```
// form the request
startBackfillRequest := request.NewStartMatchBackfill()
startBackfillRequest.RequestID = "1111aaaa-22bb-33cc-44dd-5555eeee66ff"          //
 optional
startBackfillRequest.MatchmakingConfigurationArn = "arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig"
var matchMaker model.MatchmakerData
if err := matchMaker.UnmarshalJSON([]byte(gameSession.MatchmakerData)); err != nil {

    return
}
startBackfillRequest.Players = matchMaker.Players
res, err := server.StartMatchBackfill(startBackfillRequest)

// Implement callback function for backfill
func OnUpdateGameSession(myGameSession model.GameSession) {
    // game-specific tasks to prepare for the newly matched players and update
 matchmaker data as needed
}
```

StopMatchBackfill()

Annulla una richiesta di match backfill attiva. Per ulteriori informazioni, consulta la funzione di FlexMatchriempimento.

Sintassi

```
func StopMatchBackfill(req request.StopMatchBackfillRequest) error
```

Parametri

StopMatchBackfillRequest

Un StopMatchBackfillRequest oggetto che identifica il ticket di matchmaking da annullare:

- L'ID del ticket assegnato alla richiesta di riempimento.
- Il matchmaker a cui è stata inviata la richiesta di riempimento.
- La sessione di gioco associata alla richiesta di riempimento.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
stopBackfillRequest := request.NewStopMatchBackfill()  // Use this function to create
 request
stopBackfillRequest.TicketID = "1111aaaa-22bb-33cc-44dd-5555eeee66ff"
stopBackfillRequest.MatchmakingConfigurationArn = "arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig"

//error
err := server.StopMatchBackfill(stopBackfillRequest)
```

GetComputeCertificate()

Recupera il percorso del certificato TLS utilizzato per crittografare la connessione di rete tra il server di gioco e il client di gioco. Puoi utilizzare il percorso del certificato quando registri il tuo dispositivo di elaborazione in una flotta Anywhere. Amazon GameLift Servers Per ulteriori informazioni, consulta RegisterCompute.

Sintassi

```
func GetComputeCertificate() (result.GetComputeCertificateResult, error)
```

Valore restituito

Restituisce un `GetComputeCertificateResult` oggetto che contiene quanto segue:

- CertificatePath: il percorso del certificato TLS sulla risorsa di calcolo. Quando si utilizza una flotta Amazon GameLift Servers gestita, questo percorso contiene:
  - `certificate.pem`: Il certificato dell'utente finale. La catena completa di certificati è la combinazione dei certificati `certificateChain.pem` aggiunti a questo certificato.
  - `certificateChain.pem`: La catena di certificati che contiene il certificato principale e i certificati intermedi.
  - `rootCertificate.pem`: Il certificato principale.
  - `privateKey.pem`: la chiave privata per il certificato dell'utente finale.
- ComputeName: il nome della risorsa di calcolo.

Esempio

```
tlsCertificate, err := server.GetFleetRoleCredentials(getFleetRoleCredentialsRequest)
```

GetFleetRoleCredentials()

Recupera le credenziali del ruolo di servizio che crei per estendere le autorizzazioni agli altri utenti. Servizi AWS Amazon GameLift Servers Queste credenziali consentono al server di gioco di utilizzare le tue risorse. AWS Per ulteriori informazioni, consulta Configurare un ruolo di servizio IAM per Amazon GameLift Servers.

Sintassi

```
func GetFleetRoleCredentials(
  req request.GetFleetRoleCredentialsRequest,
) (result.GetFleetRoleCredentialsResult, error) {
  return srv.getFleetRoleCredentials(&req)
}
```

Parametri

GetFleetRoleCredentialsRequest

Credenziali di ruolo che estendono l'accesso limitato alle tue AWS risorse al server di gioco.

Valore restituito

Restituisce un `GetFleetRoleCredentialsResult` oggetto che contiene quanto segue:

- AssumedRoleUserArn - L'Amazon Resource Name (ARN) dell'utente a cui appartiene il ruolo di servizio.
- AssumedRoleId - L'ID dell'utente a cui appartiene il ruolo di servizio.
- AccessKeyId - L'ID della chiave di accesso per autenticare e fornire l'accesso alle AWS risorse.
- SecretAccessKey - L'ID della chiave di accesso segreta per l'autenticazione.
- SessionToken - Un token per identificare la sessione attiva corrente che interagisce con AWS le tue risorse.
- Scadenza: il periodo di tempo che manca alla scadenza delle credenziali della sessione.

Esempio

```
// form the customer credentials request
getFleetRoleCredentialsRequest := request.NewGetFleetRoleCredentials()
getFleetRoleCredentialsRequest.RoleArn = "arn:aws:iam::123456789012:role/service-role/
exampleGameLiftAction"

credentials, err := server.GetFleetRoleCredentials(getFleetRoleCredentialsRequest)
```

Distruggi ()

Libera l'SDK del server di Amazon GameLift Servers gioco dalla memoria. È consigliabile chiamare questo metodo dopo `ProcessEnding()` e prima di terminare il processo. Se utilizzi una flotta Anywhere e non interrompi i processi del server dopo ogni sessione di gioco, chiama `Destroy()` e poi `InitSDK()` reinizializza prima di notificare Amazon GameLift Servers che il processo è pronto per ospitare una sessione di gioco. `ProcessReady()`

Sintassi

```
func Destroy() error {
 return srv.destroy()
}
```

Valore restituito

Restituisce un errore con un messaggio di errore se il metodo fallisce.

Esempio

```
// operations to end game sessions and the server process
defer func() {
  err := server.ProcessEnding()
  server.Destroy()
  if err != nil {
    fmt.Println("ProcessEnding() failed. Error: ", err)
    os.Exit(-1)
  } else {
    os.Exit(0)
  }
}
```

Server C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers

Usa il riferimento Amazon GameLift Servers Unreal Server SDK 5.x per aiutarti a preparare il tuo gioco multiplayer da utilizzare con. Amazon GameLift Servers Per dettagli sul processo di integrazione, consulta. Aggiungi Amazon GameLift Servers al tuo server di gioco Se stai usando il Amazon GameLift Servers plugin per Unreal, vedi anche Amazon GameLift Serversplugin per Unreal Engine.

> **Note**
>
> Questo argomento descrive l'API Amazon GameLift Servers C++ che puoi usare quando compili per Unreal Engine. In particolare, questa documentazione si applica al codice compilato con l'opzione. `-DBUILD_FOR_UNREAL=1`

Server C++ (Unreal) SDK 5.x per Amazon GameLift Servers -- Tipi di dati

Usa il riferimento all'Amazon GameLift ServersUnreal Server SDK 5.x per aiutarti a preparare il tuo gioco multiplayer da utilizzare con. Amazon GameLift Servers Per i dettagli sul processo di integrazione, consulta. Aggiungi Amazon GameLift Servers al tuo server di gioco Se stai usando il Amazon GameLift Servers plugin per Unreal, vedi anche Amazon GameLift Serversplugin per Unreal Engine.

> **Note**
>
> Questo argomento descrive l'API Amazon GameLift Servers C++ che puoi usare quando compili per Unreal Engine. In particolare, questa documentazione si applica al codice compilato con l'opzione. `-DBUILD_FOR_UNREAL=1`

Server C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers

Tipi di dati

- FProcessParametri
- UpdateGameSession
- GameSession
- FServerParametri

- FStartMatchBackfillRequest
- FPlayer
- FGameLiftDescribePlayerSessionsRequest
- FStopMatchBackfillRequest
- FAttributeValore
- FGameLiftGetFleetRoleCredentialsRequest
- FGameLiftLongOutcome
- FGameLiftStringOutcome
- FGameLiftDescribePlayerSessionsOutcome
- FGameLiftDescribePlayerSessionsResult
- FGenericRisultato
- FGameLiftPlayerSession
- FGameLiftGetComputeCertificateOutcome
- FGameLiftGetComputeCertificateResult
- FGameLiftGetFleetRoleCredentialsOutcome
- FGetFleetRoleCredentialsResult
- FGameLiftError
- enumerazioni;

FProcessParametri

Questo tipo di dati contiene l'insieme di parametri inviati a Amazon GameLift Servers in aProcessReady().

Proprietà	Descrizione
LogParameters	Un oggetto con percorsi di directory ai file generati durante una sessione di gioco. Amazon GameLift Serverscopia e archivia i file per accessi futuri. Tipo: `TArray<FString>` Required: No

OnHealthCheck	La funzione di callback che Amazon GameLift Servers richiama per richiedere un rapporto sullo stato di salute del processo del server. Amazon GameLift Serverschiama questa funzione ogni 60 secondi e attende una risposta per 60 secondi. Il processo del server ritorna `TRUE` se integro, `FALSE` se non integro. Se non viene restituita alcuna risposta, Amazon GameLift Servers registra il processo del server come non integro. Questa proprietà è una funzione delegata definita come`DECLARE_DELEGATE_RetVal(bool, FOnHealthCheck)  ;` Tipo: `FOnHealthCheck` Required: No
OnProcessTerminate	La funzione di callback che Amazon GameLift Servers richiama per forzare l'arresto del processo server. Dopo aver chiamato questa funzione, Amazon GameLift Servers attende 5 minuti che il processo server si spenga e risponda con una ProcessEnding() chiamata prima di chiudere il processo server. Tipo: `FSimpleDelegate` Campo obbligatorio: sì

OnStartGameSession	La funzione di callback che Amazon GameLift Servers richiama l'attivazione di una nuova sessione di gioco. Amazon GameLift Serversch iama questa funzione in risposta a una richiesta del client. CreateGameSession La funzione di callback passa un GameSession oggetto. Questa proprietà è una funzione delegata definita come `DECLARE_DELEGATE_O neParam(FOnStartGameSession , Aws::GameLift::Server::Mode l::GameSession);` Tipo: `FOnStartGameSession` Campo obbligatorio: sì
OnUpdateGameSession	La funzione di callback che Amazon GameLift Servers richiama per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift Serverschiama questa funzione quando è stata elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un GameSession oggetto, uno status update (`updateReason`) e l'ID del ticket match backfill. Questa proprietà è una funzione delegata definita come `DECLARE_DELEGATE_O neParam(FOnUpdateGameSessio n, Aws::GameLift::Server::Mode l::UpdateGameSession);` Tipo: `FOnUpdateGameSession` Required: No

Porta	Il numero di porta su cui il processo server ascolta le connessioni di nuovi giocatori . Il valore deve rientrare nella gamma di porte configurate per qualsiasi parco istanze che distribuisce questa build del server di gioco. Questo numero di porta è incluso nella sessione di gioco e negli oggetti della sessione del giocatore utilizzati dalle sessioni di gioco per la connessione a un processo server. Tipo: `int` Campo obbligatorio: sì

UpdateGameSession

Questo tipo di dati viene aggiornato a un oggetto della sessione di gioco, che include il motivo per cui la sessione di gioco è stata aggiornata e il relativo ID del ticket di backfill se il backfill viene utilizzato per riempire le sessioni dei giocatori nella sessione di gioco.

Proprietà	Descrizione
GameSession	Oggetto GameSession. L'`GameSessi on` oggetto contiene proprietà che descrivono una sessione di gioco. Tipo: `Aws::GameLift::Server::Game Session` Required: No
UpdateReason	Il motivo per cui la sessione di gioco viene aggiornata. Tipo: `enum class UpdateReason` • MATCHMAKING_DATA_UPDATED • BACKFILL_FAILED

Proprietà	Descrizione
	• BACKFILL_TIMED_OUT • BACKFILL_ANNULLATO Required: No
BackfillTicketId	L'ID del ticket di backfill che tenta di aggiornare la sessione di gioco. Tipo: `char[]` Required: No

GameSession

Questo tipo di dati fornisce i dettagli di una sessione di gioco.

Proprietà	Descrizione
GameSessionId	Un identificatore univoco per la sessione di gioco. L'ARN di una sessione di gioco ha il seguente formato:. `arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token>` Tipo: `char[]` Required: No
Nome	Un'etichetta descrittiva della sessione di gioco. Tipo: `char[]` Required: No
FleetId	Un identificatore univoco per la flotta su cui è in esecuzione la sessione di gioco.

Proprietà	Descrizione
	Tipo: `char[]` Required: No
MaximumPlayerSessionCount	Il numero massimo di connessioni dei giocatori alla sessione di gioco. Tipo: `int` Required: No
Porta	Il numero di porta per la sessione di gioco. Per connettersi a un server di Amazon GameLift Servers gioco, un'app necessita sia dell'indi rizzo IP che del numero di porta. Tipo: `int` Required: No
IpAddress	L'indirizzo IP della sessione di gioco. Per connettersi a un server di Amazon GameLift Servers gioco, un'app necessita sia dell'indi rizzo IP che del numero di porta. Tipo: `char[]` Required: No
GameSessionData	Un set di proprietà personalizzate della sessione di gioco, formattate come valore di stringa singola. Tipo: `char[]` Required: No

Proprietà	Descrizione
MatchmakerData	Informazioni sul processo di matchmaking utilizzato per creare la sessione di gioco, in sintassi JSON, formattate come stringa. Oltre alla configurazione di matchmaking utilizzata, contiene dati su tutti i giocatori assegnati alla partita, compresi gli attributi dei giocatori e le assegnazioni delle squadre. Tipo: `char[]` Required: No
GameProperties	Un insieme di proprietà personalizzate per una sessione di gioco, formattate come coppie chiave:valore. Queste proprietà vengono passate con una richiesta di avvio di una nuova sessione di gioco. Tipo: `GameProperty[]` Required: No

Proprietà	Descrizione
DnsName	L'identificatore DNS assegnato all'istanza che esegue la sessione di gioco. I valori hanno il seguente formato: • Flotte abilitate per TLS:. `<unique identifier>.<region identifier>.amazongamelift.com` • Non-TLS-enabled flotte:. `ec2-<unique identifier>.compute.amazonaws.com` Quando ti connetti a una sessione di gioco in esecuzione su una flotta abilitata per TLS, devi usare il nome DNS, non l'indirizzo IP. Tipo: `char[]` Required: No

FServerParametri

Informazioni utilizzate per mantenere la connessione tra un server Amazon GameLift Servers Anywhere e il Amazon GameLift Servers servizio. Queste informazioni vengono utilizzate quando si avviano nuovi processi server conInitSDK(). Per i server ospitati su EC2 istanze Amazon GameLift Servers gestite, utilizza un oggetto vuoto.

Proprietà	Descrizione
webSocketUrl	`GameLiftServerSdkEndpoint` Amazon GameLift ServersRestituisce quando si utilizza una `RegisterCompute` risorsa di calcolo Amazon GameLift Servers Anywhere. Tipo: `char[]`

Proprietà	Descrizione
	Campo obbligatorio: sì
ID del processo	Un identificatore univoco registrato nel processo del server che ospita il gioco. Tipo: `char[]` Campo obbligatorio: sì
hostId	`HostID`Viene `ComputeName` utilizzato al momento della registrazione del computer. Per ulteriori informazioni, consultare RegisterCompute. Tipo: `char[]` Campo obbligatorio: sì
FleetID	L'identificatore univoco della flotta su cui è registrato il computer. Per ulteriori informazioni, consultare RegisterCompute. Tipo: `char[]` Campo obbligatorio: sì
AuthToken	Il token di autenticazione generato da Amazon GameLift Servers questo autentica il server su. Amazon GameLift Servers Per ulteriori informazioni, consultare GetComputeAuthToken. Tipo: `char[]` Campo obbligatorio: sì

FStartMatchBackfillRequest

Informazioni utilizzate per creare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni durante una chiamataAmazon GameLift Servers. StartMatchBackfill()

Proprietà	Descrizione
GameSessionArn	Un identificatore univoco della sessione di gioco. L'operazione API `GetGameSessionId` restituisce l'identificatore in formato ARN. Tipo: `char[]` Campo obbligatorio: sì
MatchmakingConfigurationArn	Un identificatore univoco, sotto forma di ARN, che il matchmaker può utilizzare per questa richiesta. L'ARN del matchmaker per la sessione di gioco originale si trova nell'oggetto della sessione di gioco nella proprietà matchmaker data. Scopri di più sui dati dei matchmaker in Work with matchmaker data. Tipo: `char[]` Campo obbligatorio: sì
Players	Un insieme di dati che rappresenta tutti i giocatori che partecipano alla sessione di gioco. Il matchmaker utilizza queste informazioni per cercare nuovi giocatori che rappresentano un buon abbinamento per i giocatori attuali. Tipo: `TArray<FPlayer>` Campo obbligatorio: sì
TicketId	Un identificatore univoco per un ticket di richiesta di matchmaking o match backfill.

Proprietà	Descrizione
	Se non fornisci un valore, Amazon GameLift Servers ne genera uno. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill degli abbinamenti o annullare la richiesta, se necessario. Tipo: `char[]` Required: No

FPlayer

Questo tipo di dati rappresenta un giocatore in matchmaking. Quando si avvia una richiesta di matchmaking, un giocatore ha un ID giocatore, attributi e possibilmente dati di latenza. Amazon GameLift Serversaggiunge le informazioni sulla squadra dopo una partita.

Proprietà	Descrizione
LatencyInSM	Un insieme di valori espressi in milliseco ndi che indica la quantità di latenza che un giocatore sperimenta quando è connesso a una posizione. Se viene utilizzata questa proprietà, il giocatore viene abbinato solo per le posizioni elencate. Se un matchmaker ha una regola che valuta la latenza dei giocatori, i giocatori devono segnalare la latenza per essere abbinati. Tipo: `TMap>FString, int32<` Required: No
PlayerAttributes	Una raccolta di coppie chiave:valore contenent i informazioni sui giocatori da utilizzare nel matchmaking. Le chiavi degli attributi del giocatore devono corrispondere a quelle

Proprietà	Descrizione
	PlayerAttributes utilizzate in un set di regole di matchmaking. Per ulteriori informazioni sugli attributi dei giocatori, vedi AttributeValue. Tipo: `TMap>FString, FAttribut eValue<` Required: No
PlayerId	Un identificatore univoco per un giocatore. Tipo: `std::string` Required: No
Team	Il nome della squadra a cui è assegnato il giocatore in una partita. Definisci il nome della squadra nel set di regole del matchmaking. Tipo: `FString` Required: No

FGameLiftDescribePlayerSessionsRequest

Un oggetto che specifica le sessioni dei giocatori da recuperare. Il processo server fornisce queste informazioni con una DescribePlayerSessions() chiamata a. Amazon GameLift Servers

Proprietà	Descrizione
GameSessionId	Un identificatore univoco della sessione di gioco. Utilizzare questo parametro per richieder e tutte le sessioni giocatore per la sessione di gioco specificata.

Proprietà	Descrizione
	Il formato dell'ID della sessione di gioco è`FString`. `GameSessionID` È una stringa ID personalizzata o un Tipo: `std::string` Required: No
PlayerSessionId	L'identificatore univoco per la sessione di un giocatore. Usa questo parametro per richiedere una singola sessione di gioco specifica. Tipo: `FString` Required: No
PlayerId	L'identificatore univoco di un giocatore. Usa questo parametro per richiedere tutte le sessioni di gioco per un giocatore specifico. Consultare Genera giocatore IDs. Tipo: `FString` Required: No

Proprietà	Descrizione
PlayerSessionStatusFilter	Lo stato della sessione del giocatore su cui filtrare i risultati. I possibili stati delle sessioni di gioco includono: • RISERVATO — La richiesta di sessione del giocatore è stata ricevuta, ma il giocatore non si è connesso al processo del server né è stato convalidato. • ATTIVO: il giocatore è stato convalidato dal processo del server ed è connesso. • COMPLETATA — La connessione del giocatore è stata interrotta. • TIMEDOUT: è stata ricevuta una richiesta di sessione, ma il giocatore non si è connesso o non è stato convalidato entro il limite di timeout (60 secondi). Tipo: `FString` Required: No
NextToken	Il token che indica l'inizio della pagina successiva di risultati. Per specificare l'inizio del set di risultati, non fornire un valore. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `FString` Required: No

Proprietà	Descrizione
Limite	Numero massimo di risultati da restituire. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `int` Required: No

FStopMatchBackfillRequest

Informazioni utilizzate per annullare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni al Amazon GameLift Servers servizio durante una chiamata. StopMatchBackfill()

Proprietà	Descrizione
GameSessionArn	Un identificatore univoco della sessione di gioco della richiesta annullata. Tipo: `FString` Campo obbligatorio: sì
MatchmakingConfigurationArn	Un identificatore univoco del matchmaker a cui è stata inviata questa richiesta. Tipo: `FString` Campo obbligatorio: sì
TicketId	Un identificatore univoco del ticket di richiesta di riempimento da annullare. Tipo: `FString` Campo obbligatorio: sì

FAttributeValore

Utilizzate questi valori nelle coppie chiave-valore FPlayer degli attributi. Questo oggetto consente di specificare il valore di un attributo utilizzando uno qualsiasi dei tipi di dati validi: stringa, numero, array di stringhe o mappa di dati. Ogni `AttributeValue` oggetto può utilizzare solo una delle proprietà disponibili.

Proprietà	Descrizione
AttrType	Speciifica il tipo di valore dell'attributo. Tipo: un valore `FAttributeType` enum. Required: No
S	Rappresenta un valore di attributo di tipo stringa. Tipo: `FString` Required: No
N	Rappresenta un valore di attributo numerico. Tipo: `double` Required: No
SL	Rappresenta una matrice di valori di attributi di tipo stringa. Tipo: `TArray<FString>` Required: No
SDM	Rappresenta un dizionario di chiavi di stringa e valori doppi. Tipo: `TMap<FString, double>` Required: No

FGameLiftGetFleetRoleCredentialsRequest

Questo tipo di dati fornisce credenziali di ruolo che estendono l'accesso limitato alle tue AWS risorse al server di gioco. Per ulteriori informazioni, consultare Configurare un ruolo di servizio IAM per Amazon GameLift Servers.

Proprietà	Descrizione
RoleArn	L'Amazon Resource Name (ARN) del ruolo di servizio che estende l'accesso limitato alle tue AWS risorse. Tipo: `FString` Required: No
RoleSessionName	Il nome della sessione che descrive l'uso delle credenziali del ruolo. Tipo: `FString` Required: No

FGameLiftLongOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: `long` Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `long&&`

Proprietà	Descrizione
	Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “FGameLiftError” Required: No

FGameLiftStringOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: `FString` Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `FString&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo.

Proprietà	Descrizione
	Tipo: the section called “FGameLiftError” Required: No

FGameLiftDescribePlayerSessionsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “FGameLiftDescribePlayerSessionsResult” Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `FGameLiftDescribePlayerSessionsResult&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “FGameLiftError” Required: No

FGameLiftDescribePlayerSessionsResult

Proprietà	Descrizione
PlayerSessions	Tipo: `TArray<FGameLiftPlayerSession>` Campo obbligatorio: sì
NextToken	Il token che indica l'inizio della pagina successiva dei risultati. Per specificare l'inizio del set di risultati, non fornire un valore. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato. Tipo: `FString` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “FGameLiftError” Required: No

FGenericRisultato

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool`

Proprietà	Descrizione
	Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “FGameLiftError” Required: No

FGameLiftPlayerSession

Proprietà	Descrizione
CreationTime	Tipo: `long` Campo obbligatorio: sì
FleetId	Tipo: `FString` Campo obbligatorio: sì
GameSessionId	Tipo: `FString` Campo obbligatorio: sì
IpAddress	Tipo: `FString` Campo obbligatorio: sì
PlayerData	Tipo: `FString` Campo obbligatorio: sì
PlayerId	Tipo: `FString` Campo obbligatorio: sì
PlayerSessionId	Tipo: `FString` Campo obbligatorio: sì

Proprietà	Descrizione
Porta	Tipo: `int` Campo obbligatorio: sì
Stato	Tipo: Un `PlayerSessionStatus` enum. Campo obbligatorio: sì
TerminationTime	Tipo: `long` Campo obbligatorio: sì
DnsName	Tipo: `FString` Campo obbligatorio: sì

FGameLiftGetComputeCertificateOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “FGameLiftGetComputeCertificateResult” Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: `FGameLiftGetComputeCertificateResult&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool`

Proprietà	Descrizione
	Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “FGameLiftError” Required: No

FGameLiftGetComputeCertificateResult

Il percorso del certificato TLS sul computer e il nome host del calcolo.

Proprietà	Descrizione
CertificatePath	Tipo: `FString` Campo obbligatorio: sì
ComputeName	Tipo: `FString` Campo obbligatorio: sì

FGameLiftGetFleetRoleCredentialsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	Il risultato dell'azione. Tipo: the section called “FGetFleetRoleCredentialsResult” Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.

Proprietà	Descrizione
	Tipo: `FGameLiftGetFleetRoleCredentialsResult&&` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “FGameLiftError” Required: No

FGetFleetRoleCredentialsResult

Proprietà	Descrizione
AccessKeyId	L'ID della chiave di accesso per autenticare e fornire l'accesso alle AWS risorse. Tipo: `FString` Required: No
AssumedRoleId	L'ID dell'utente a cui appartiene il ruolo di servizio. Tipo: `FString` Required: No
AssumedRoleUserArn	L'Amazon Resource Name (ARN) dell'utente a cui appartiene il ruolo di servizio. Tipo: `FString`

Proprietà	Descrizione
	Required: No
Scadenza	Il periodo di tempo che manca alla scadenza delle credenziali di sessione. Tipo: `FDateTime` Required: No
SecretAccessKey	L'ID della chiave di accesso segreta per l'autenticazione. Tipo: `FString` Required: No
SessionToken	Un token per identificare la sessione attiva corrente che interagisce con AWS le tue risorse. Tipo: `FString` Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: `bool` Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “GameLiftError” Required: No

FGameLiftError

Proprietà	Descrizione
ErrorType	Il tipo di errore.

Proprietà	Descrizione
	Tipo: Un `GameLiftErrorType` enum. Required: No
ErrorName	Il nome dell'errore. Tipo: `std::string` Required: No
ErrorMessage	Messaggio di errore. Tipo: `std::string` Required: No

enumerazioni;

Le enumerazioni definite per l'SDK del server per Amazon GameLift Servers (Unreal) sono definite come segue:

FAttributeTipo

- NONE
- STRINGA
- DOPPIA
- ELENCO_STRINGHE
- STRING_DOUBLE_MAP

GameLiftErrorType

Valore di stringa che indica il tipo di errore. I valori validi includono:

- SERVICE_CALL_FAILED — Una chiamata a un servizio non è riuscita. AWS
- LOCAL_CONNECTION_FAILED — La connessione locale a non è riuscita. Amazon GameLift Servers
- NETWORK_NOT_INITIALIZED — La rete non è stata inizializzata.
- GAMESESSION_ID_NOT_SET — L'ID della sessione di gioco non è stato impostato.

- BAD_REQUEST_EXCEPTION
- ECCEZIONE_DI_SERVIZIO INTERNA
- ALREADY_INITIALIZED — Il Amazon GameLift Servers server o il client è già stato inizializzato con Initialize ().
- FLEET_MISMATCH — La flotta bersaglio non corrisponde alla flotta di una GameSession o PlayerSession.
- GAMELIFT_CLIENT_NOT_INITIALIZED — Il client non è stato inizializzato. Amazon GameLift Servers
- GAMELIFT_SERVER_NOT_INITIALIZED — Amazon GameLift Servers Il server non è stato inizializzato.
- GAME_SESSION_ENDED_FAILED: l'SDK del server non è riuscito a contattare il servizio per segnalare la fine della sessione di gioco. Amazon GameLift Servers
- GAME_SESSION_NOT_READY — La sessione di gioco sul server non è stata attivata. Amazon GameLift Servers
- GAME_SESSION_READY_FAILED: l'SDK del server per cui non è stato possibile contattare il servizio per segnalare che la sessione di gioco è pronta. Amazon GameLift Servers
- INITIALIZATION_MISMATCH — È stato chiamato un metodo client dopo Server: :Initialize () o viceversa.
- NOT_INITIALIZED — Il Amazon GameLift Servers server o il client non sono stati inizializzati con Initialize ().
- NO_TARGET_ALIASID_SET — Non è stato impostato un aliasID di destinazione.
- NO_TARGET_FLEET_SET — Non è stata impostata una flotta di obiettivi.
- PROCESS_ENDING_FAILED: l'SDK del server non è riuscito a contattare il servizio per segnalare la fine del processo. Amazon GameLift Servers
- PROCESS_NOT_ACTIVE — Il processo del server non è ancora attivo, non è associato a un processo e non può accettarlo o elaborarlo. GameSession PlayerSessions
- PROCESS_NOT_READY — Il processo del server non è ancora pronto per essere attivato.
- PROCESS_READY_FAILED: l'SDK del server non è riuscito a contattare il servizio per Amazon GameLift Servers segnalare che il processo è pronto.
- SDK_VERSION_DETECTION_FAILED — Rilevamento della versione SDK non riuscito.
- STX_CALL_FAILED — Una chiamata al componente di backend del server non è riuscita. XStx
- STX_INITIALIZATION_FAILED — L'inizializzazione del componente di backend del server non è riuscita. XStx

- UNEXPECTED_PLAYER_SESSION — Il server ha rilevato una sessione di giocatore non registrata.
- WEBSOCKET_CONNECT_FAILURE
- WEBSOCKET_CONNECT_FAILURE_FORBIDDEN
- URL NON VALIDO DI WEBSOCKET_CONNECT_FAILURE_INVALID_
- WEBSOCKET_CONNECT_FAILURE_TIMEOUT
- WEBSOCKET_RETRIABLE_SEND_MESSAGE_FAILURE — Errore recuperabile nell'invio di un messaggio al servizio. GameLift WebSocket
- WEBSOCKET_SEND_MESSAGE_FAILURE — Errore nell'invio di un messaggio al GameLift servizio. WebSocket
- MATCH_BACKFILL_REQUEST_VALIDATION — La convalida della richiesta non è riuscita.
- PLAYER_SESSION_REQUEST_VALIDATION — La convalida della richiesta non è riuscita.

EPlayerSessionCreationPolicy

Valore della stringa che indica se la sessione di gioco accetta nuovi giocatori. I valori validi includono:

- ACCEPT_ALL: accetta tutte le nuove sessioni giocatore.
- DENY_ALL: rifiuta tutte le nuove sessioni giocatore.
- NOT_SET — La sessione di gioco non è impostata per accettare o rifiutare sessioni di nuovi giocatori.

EPlayerSessionStatus

- ACTIVE
- COMPLETED
- NOT_SET
- RISERVATO
- TIMEOUT

Server C++ (Unreal) SDK 5.x per Amazon GameLift Servers -- Tipi di dati

Argomenti

- GetSdkVersion()
- InitSDK()

- InitSDK()
- ProcessReady()
- ProcessEnding()
- ActivateGameSession()
- UpdatePlayerSessionCreationPolicy()
- GetGameSessionId()
- GetTerminationTime()
- AcceptPlayerSession()
- RemovePlayerSession()
- DescribePlayerSessions()
- StartMatchBackfill()
- StopMatchBackfill()
- GetComputeCertificate()
- GetFleetRoleCredentials()

GetSdkVersion()

Restituisce il numero di versione corrente dell'SDK integrato nel processo del server.

Sintassi

```
FGameLiftStringOutcome GetSdkVersion();
```

Valore restituito

Se l'esito è positivo, restituisce la versione corrente dell'SDK come oggetto the section called "FGameLiftStringOutcome". L'oggetto restituito include il numero di versione (esempio`5.0.0`). Se l'esito è negativo, verrà restituito un messaggio di errore.

Esempio

```
Aws::GameLift::AwsStringOutcome SdkVersionOutcome =
 Aws::GameLift::Server::GetSdkVersion();
```

InitSDK()

Inizializza l'Amazon GameLift ServersSDK per una flotta gestita. EC2 Richiama questo metodo all'avvio, prima che si verifichi qualsiasi altra inizializzazione correlata a. Amazon GameLift Servers Questo metodo legge i parametri del server dall'ambiente host per configurare la comunicazione tra il server e il Amazon GameLift Servers servizio.

Sintassi

```
FGameLiftGenericOutcome InitSDK()
```

Valore restituito

In caso di successo, restituisce un `InitSdkOutcome` oggetto che indica che il processo del server è pronto per la chiamataProcessReady().

Esempio

```
//Call InitSDK to establish a local connection with the Amazon GameLift Servers Agent
 to enable further communication.
FGameLiftGenericOutcome initSdkOutcome = gameLiftSdkModule->InitSDK();
```

InitSDK()

Inizializza l'Amazon GameLift ServersSDK per una flotta Anywhere o una flotta di container gestita. Richiama questo metodo all'avvio, prima che si verifichi qualsiasi altra inizializzazione correlata a. Amazon GameLift Servers Questo metodo richiede parametri espliciti del server per configurare la comunicazione tra il server e il Amazon GameLift Servers servizio.

Sintassi

```
FGameLiftGenericOutcome InitSDK(serverParameters)
```

Parametri

FServerParametri

Per inizializzare un server di gioco su una flotta Amazon GameLift Servers Anywhere, costruisci un `ServerParameters` oggetto con le seguenti informazioni:

- L'URL WebSocket utilizzato per connettersi al server di gioco.
- L'ID del processo utilizzato per ospitare il server di gioco.
- L'ID del computer che ospita i processi del server di gioco.
- L'ID della Amazon GameLift Servers flotta contenente il tuo computer Amazon GameLift Servers Anywhere.
- Il token di autorizzazione generato dall'Amazon GameLift Serversoperazione.

Valore restituito

In caso di successo, restituisce un `InitSdkOutcome` oggetto che indica che il processo del server è pronto per la chiamataProcessReady().

> **Note**
>
> Se le chiamate a non `InitSDK()` riescono per le build di gioco distribuite sulle flotte Anywhere, controlla il `ServerSdkVersion` parametro utilizzato durante la creazione della risorsa di compilazione. È necessario impostare esplicitamente questo valore sulla versione SDK del server in uso. Il valore predefinito per questo parametro è 4.x, che non è compatibile. Per risolvere questo problema, crea una nuova build e distribuiscila in una nuova flotta.

Esempio

```
//Define the server parameters
FServerParameters serverParameters;
parameters.m_authToken = "1111aaaa-22bb-33cc-44dd-5555eeee66ff";
parameters.m_fleetId = "arn:aws:gamelift:us-west-1:111122223333:fleet/
fleet-9999ffff-88ee-77dd-66cc-5555bbbb44aa";
parameters.m_hostId = "HardwareAnywhere";
parameters.m_processId = "PID1234";
parameters.m_webSocketUrl = "wss://us-west-1.api.amazongamelift.com";

//Call InitSDK to establish a local connection with the Amazon GameLift Servers Agent
 to enable further communication.
FGameLiftGenericOutcome initSdkOutcome = gameLiftSdkModule->InitSDK(serverParameters);
```

ProcessReady()

Notifica Amazon GameLift Servers che il processo del server è pronto per ospitare sessioni di gioco. Chiama questo metodo dopo averlo InitSDK() invocato. Questo metodo deve essere chiamato solo una volta per processo.

Sintassi

```
GenericOutcome ProcessReady(const Aws::GameLift::Server::ProcessParameters
&processParameters);
```

Parametri

processParameters

Un FProcessParametri oggetto che comunica le seguenti informazioni sul processo del server:

- Nomi dei metodi di callback implementati nel codice del server di gioco richiamato dal Amazon GameLift Servers servizio per comunicare con il processo server.
- Numero di porta sulla quale è in ascolto il processo del server.
- Percorso verso qualsiasi file specifico di una sessione di gioco che Amazon GameLift Servers deve acquisire e archiviare.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra la chiamata ProcessReady() e le implementazioni della funzione delegata.

```
//Calling ProcessReady tells Amazon GameLift Servers this game server is ready to
 receive incoming game sessions!
UE_LOG(GameServerLog, Log, TEXT("Calling Process Ready"));
FGameLiftGenericOutcome processReadyOutcome = gameLiftSdkModule-
>ProcessReady(*params);
```

ProcessEnding()

Notifica Amazon GameLift Servers che il processo del server sta terminando. Chiama questo metodo dopo tutte le altre attività di pulizia (inclusa la chiusura della sessione di gioco attiva) e prima di terminare il processo. A seconda del risultato di`ProcessEnding()`, il processo termina con

successo (0) o errore (-1) e genera un evento relativo alla flotta. Se il processo termina con un errore, l'evento di flotta generato è`SERVER_PROCESS_TERMINATED_UNHEALTHY`).

Sintassi

```
FGameLiftGenericOutcome ProcessEnding()
```

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

```
//OnProcessTerminate callback. Amazon GameLift Servers will invoke this callback before
 shutting down an instance hosting this game server.
//It gives this game server a chance to save its state, communicate with services,
 etc., before being shut down.
//In this case, we simply tell Amazon GameLift Servers we are indeed going to shutdown.
params->OnTerminate.BindLambda([=]() {
  UE_LOG(GameServerLog, Log, TEXT("Game Server Process is terminating"));
  gameLiftSdkModule->ProcessEnding();
});
```

ActivateGameSession()

Notifica Amazon GameLift Servers che il processo del server ha attivato una sessione di gioco ed è ora pronto a ricevere le connessioni dei giocatori. Questa azione dovrebbe essere richiamata come parte della funzione di on`StartGameSession()` callback, dopo l'inizializzazione di tutta la sessione di gioco.

Sintassi

```
FGameLiftGenericOutcome ActivateGameSession()
```

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio mostra le `ActivateGameSession()` chiamate come parte della funzione on`StartGameSession()` delegata.

```
//When a game session is created, Amazon GameLift Servers sends an activation request
 to the game server and passes along the game session object containing game properties
 and other settings.
//Here is where a game server should take action based on the game session object.
//Once the game server is ready to receive incoming player connections, it should
 invoke GameLiftServerAPI.ActivateGameSession()
auto onGameSession = [=](Aws::GameLift::Server::Model::GameSession gameSession)
{
  FString gameSessionId = FString(gameSession.GetGameSessionId());
  UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"), *gameSessionId);
  gameLiftSdkModule->ActivateGameSession();
};
```

UpdatePlayerSessionCreationPolicy()

Aggiorna la capacità della sessione di gioco corrente di accettare nuove sessioni giocatore. Una sessione di gioco può essere configurata per accettare o rifiutare tutte le nuove sessioni giocatore.

Sintassi

```
FGameLiftGenericOutcome UpdatePlayerSessionCreationPolicy(EPlayerSessionCreationPolicy
 policy)
```

Parametri

playerCreationSessionPolitica

Valore della stringa che indica se la sessione di gioco accetta nuovi giocatori.

I valori validi includono:

- ACCEPT_ALL: accetta tutte le nuove sessioni giocatore.
- DENY_ALL: rifiuta tutte le nuove sessioni giocatore.

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio definisce la policy di partecipazione alla sessione di gioco corrente per accettare tutti i giocatori.

```
FGameLiftGenericOutcome outcome = gameLiftSdkModule-
>UpdatePlayerSessionCreationPolicy(Aws::GameLift::Model::EPlayerSessionCreationPolicy::ACCEPT_A
```

GetGameSessionId()

Recupera l'ID della sessione di gioco ospitata dal processo del server attivo.

Per i processi inattivi che non vengono attivati con una sessione di gioco, la chiamata restituisce un. the section called “FGameLiftError”

Sintassi

```
FGameLiftStringOutcome GetGameSessionId()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Se l'esito è positivo, l'ID della sessione di gioco verrà restituito come oggetto the section called “FGameLiftStringOutcome”. Se non riesce, restituisce un messaggio di errore».

Per i processi inattivi che non vengono attivati con una sessione di gioco, la chiamata restituisce `Success` = `True` e `GameSessionId` ="".

Esempio

```
//When a game session is created, Amazon GameLift Servers sends an activation request
 to the game server and passes along the game session object containing game properties
 and other settings.
//Here is where a game server should take action based on the game session object.
//Once the game server is ready to receive incoming player connections, it should
 invoke GameLiftServerAPI.ActivateGameSession()
auto onGameSession = [=](Aws::GameLift::Server::Model::GameSession gameSession)
{
  FString gameSessionId = FString(gameSession.GetGameSessionId());
  UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"), *gameSessionId);
  gameLiftSdkModule->ActivateGameSession();
};
```

GetTerminationTime()

Restituisce il tempo di arresto pianificato di un processo del server, se è disponibile un tempo di chiusura. Un processo server interviene dopo aver ricevuto una `onProcessTerminate()` richiamata da. Amazon GameLift Servers Amazon GameLift Serverschiama `onProcessTerminate()` per i seguenti motivi:

- Quando il processo del server ha segnalato problemi di salute o non ha risposto. Amazon GameLift Servers
- Quando si termina l'istanza durante un evento di scale-down.
- Quando un'istanza viene terminata a causa di un'interruzione di un'istanza spot.

Sintassi

```
AwsDateTimeOutcome GetTerminationTime()
```

Valore restituito

In caso di successo, restituisce l'ora di terminazione come oggetto. `AwsDateTimeOutcome` Il valore è il tempo di terminazione, espresso in segni di spunta trascorsi da allora. `0001 00:00:00` Ad esempio, il valore della data e dell'ora è uguale ai segni di `2020-09-13 12:26:40 -000Z` spunta. `637355968000000000` Se non è disponibile alcun orario di terminazione, restituisce un messaggio di errore.

Se il processo non ha ricevuto una `ProcessParameters.OnProcessTerminate()` richiamata, viene restituito un messaggio di errore. Per ulteriori informazioni sulla chiusura di un processo server, vedere. Rispondi a una notifica di chiusura di un processo del server

Esempio

```
AwsDateTimeOutcome TermTimeOutcome = gameLiftSdkModule->GetTerminationTime();
```

AcceptPlayerSession()

Notifica Amazon GameLift Servers che un giocatore con l'ID di sessione specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift Serversverifica che l'ID della sessione del giocatore sia valido. Dopo la convalida della sessione di gioco, Amazon GameLift Servers modifica lo stato dello slot del giocatore da RISERVATO ad ATTIVO.

Sintassi

```
FGameLiftGenericOutcome AcceptPlayerSession(const FString& playerSessionId)
```

Parametri

playerSessionId

ID univoco rilasciato Amazon GameLift Servers quando viene creata una nuova sessione giocatore.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio gestisce una richiesta di connessione che include la convalida e il rifiuto di una sessione giocatore non valida. IDs

```
bool GameLiftManager::AcceptPlayerSession(const FString& playerSessionId, const
 FString& playerId)
{
  #if WITH_GAMELIFT
  UE_LOG(GameServerLog, Log, TEXT("Accepting GameLift PlayerSession: %s . PlayerId:
 %s"), *playerSessionId, *playerId);
  FString gsId = GetCurrentGameSessionId();
  if (gsId.IsEmpty()) {
    UE_LOG(GameServerLog, Log, TEXT("No GameLift GameSessionId. Returning early!"));
    return false;
  }

  if (!gameLiftSdkModule->AcceptPlayerSession(playerSessionId).IsSuccess()) {
    UE_LOG(GameServerLog, Log, TEXT("PlayerSession not Accepted."));
    return false;
  }

  // Add PlayerSession from internal data structures keeping track of connected players
  connectedPlayerSessionIds.Add(playerSessionId);
  idToPlayerSessionMap.Add(playerSessionId, PlayerSession{ playerId,
 playerSessionId });
  return true;
```

```
  #else
  return false;
  #endif
}
```

RemovePlayerSession()

Notifica Amazon GameLift Servers che un giocatore si è disconnesso dal processo del server. In risposta, Amazon GameLift Servers modifica lo slot del giocatore rendendolo disponibile.

Sintassi

```
FGameLiftGenericOutcome RemovePlayerSession(const FString& playerSessionId)
```

Parametri

`playerSessionId`

ID univoco rilasciato Amazon GameLift Servers quando viene creata una nuova sessione giocatore.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
bool GameLiftManager::RemovePlayerSession(const FString& playerSessionId)
{
  #if WITH_GAMELIFT
  UE_LOG(GameServerLog, Log, TEXT("Removing GameLift PlayerSession: %s"),
 *playerSessionId);

  if (!gameLiftSdkModule->RemovePlayerSession(playerSessionId).IsSuccess()) {
    UE_LOG(GameServerLog, Log, TEXT("PlayerSession Removal Failed"));
    return false;
  }

  // Remove PlayerSession from internal data structures that are keeping track of
 connected players
  connectedPlayerSessionIds.Remove(playerSessionId);
```

```
  idToPlayerSessionMap.Remove(playerSessionId);

  // end the session if there are no more players connected
  if (connectedPlayerSessionIds.Num() == 0) {
    EndSession();
  }

  return true;
  #else
  return false;
  #endif
}
```

DescribePlayerSessions()

Recupera i dati della sessione del giocatore che includono impostazioni, metadati della sessione e dati del giocatore. Utilizzate questo metodo per ottenere informazioni su quanto segue:

- Una sessione per giocatore singolo
- Tutte le sessioni dei giocatori in una sessione di gioco
- Tutte le sessioni dei giocatori sono associate a un ID giocatore singolo

Sintassi

```
FGameLiftDescribePlayerSessionsOutcome DescribePlayerSessions(const
 FGameLiftDescribePlayerSessionsRequest &describePlayerSessionsRequest)
```

Parametri

FGameLiftDescribePlayerSessionsRequest

Un the section called “FGameLiftDescribePlayerSessionsRequest” oggetto che descrive le sessioni dei giocatori da recuperare.

Valore restituito

Se l'esito è positivo, restituisce un oggetto the section called “FGameLiftDescribePlayerSessionsOutcome” contenente un set di oggetti di sessione giocatore corrispondente ai parametri della richiesta.

Esempio

Questo esempio richiede che tutte le sessioni dei giocatori siano connesse attivamente a una sessione di gioco specificata. Omettendo NextTokene impostando il valore Limite su 10, Amazon GameLift Servers restituisce i primi 10 record di sessione tra giocatori corrispondenti alla richiesta.

```
void GameLiftManager::DescribePlayerSessions()
{
  #if WITH_GAMELIFT
  FString localPlayerSessions;
  for (auto& psId : connectedPlayerSessionIds)
  {
    PlayerSession ps = idToPlayerSessionMap[psId];
    localPlayerSessions += FString::Printf(TEXT("%s : %s  ; "), *(ps.playerSessionId),
 *(ps.playerId));
  }
  UE_LOG(GameServerLog, Log, TEXT("LocalPlayerSessions: %s"), *localPlayerSessions);

  UE_LOG(GameServerLog, Log, TEXT("Describing PlayerSessions in this GameSession"));
  FGameLiftDescribePlayerSessionsRequest request;
  request.m_gameSessionId = GetCurrentGameSessionId();

  FGameLiftDescribePlayerSessionsOutcome outcome = gameLiftSdkModule-
>DescribePlayerSessions(request);
  LogDescribePlayerSessionsOutcome(outcome);
  #endif
}
```

StartMatchBackfill()

Invia una richiesta per trovare nuovi giocatori per gli slot aperti in una sessione di gioco creata con FlexMatch. Per ulteriori informazioni, consulta la funzione di FlexMatchriempimento.

Questa operazione è asincrona. Se vengono abbinati nuovi giocatori, Amazon GameLift Servers fornisce dati aggiornati sul matchmaker utilizzando la funzione di callback. `OnUpdateGameSession()`

Un processo del server può avere un solo backfill degli abbinamenti attivo alla volta. Per inviare una nuova richiesta, chiama prima StopMatchBackfill() per annullare la richiesta originale.

Sintassi

```
FGameLiftStringOutcome StartMatchBackfill (FStartMatchBackfillRequest
 &startBackfillRequest);
```

Parametri

FStartMatchBackfillRequest

Un StartMatchBackfillRequest oggetto che comunica le seguenti informazioni:

- ID del ticket da assegnare alla richiesta di backfill. Queste informazioni sono facoltative; se non viene fornito alcun ID, ne Amazon GameLift Servers genererà uno.
- Matchmaker a cui inviare la richiesta. L'ARN di configurazione completo è obbligatorio. Questo valore si trova nei dati del matchmaker della sessione di gioco.
- L'ID della sessione di gioco da riempire.
- I dati di matchmaking disponibili per i giocatori attuali della sessione di gioco.

Valore restituito

Restituisce un `StartMatchBackfillOutcome` oggetto con l'ID del ticket match backfill, oppure restituisce un errore con un messaggio di errore.

Esempio

```
FGameLiftStringOutcome FGameLiftServerSDKModule::StartMatchBackfill(const
 FStartMatchBackfillRequest& request)
{
  #if WITH_GAMELIFT
  Aws::GameLift::Server::Model::StartMatchBackfillRequest sdkRequest;
  sdkRequest.SetTicketId(TCHAR_TO_UTF8(*request.m_ticketId));
  sdkRequest.SetGameSessionArn(TCHAR_TO_UTF8(*request.m_gameSessionArn));

 sdkRequest.SetMatchmakingConfigurationArn(TCHAR_TO_UTF8(*request.m_matchmakingConfigurationArn
  for (auto player : request.m_players) {
    Aws::GameLift::Server::Model::Player sdkPlayer;
    sdkPlayer.SetPlayerId(TCHAR_TO_UTF8(*player.m_playerId));
    sdkPlayer.SetTeam(TCHAR_TO_UTF8(*player.m_team));
    for (auto entry : player.m_latencyInMs) {
      sdkPlayer.WithLatencyMs(TCHAR_TO_UTF8(*entry.Key), entry.Value);
    }
```

```
   std::map<std::string, Aws::GameLift::Server::Model::AttributeValue>
sdkAttributeMap;
   for (auto attributeEntry : player.m_playerAttributes) {
     FAttributeValue value = attributeEntry.Value;
     Aws::GameLift::Server::Model::AttributeValue attribute;
     switch (value.m_type) {
       case FAttributeType::STRING:
         attribute =
Aws::GameLift::Server::Model::AttributeValue(TCHAR_TO_UTF8(*value.m_S));
       break;
       case FAttributeType::DOUBLE:
         attribute = Aws::GameLift::Server::Model::AttributeValue(value.m_N);
       break;
       case FAttributeType::STRING_LIST:
         attribute =
Aws::GameLift::Server::Model::AttributeValue::ConstructStringList();
         for (auto sl : value.m_SL) {
           attribute.AddString(TCHAR_TO_UTF8(*sl));
         };
       break;
       case FAttributeType::STRING_DOUBLE_MAP:
         attribute =
Aws::GameLift::Server::Model::AttributeValue::ConstructStringDoubleMap();
         for (auto sdm : value.m_SDM) {
           attribute.AddStringAndDouble(TCHAR_TO_UTF8(*sdm.Key), sdm.Value);
         };
       break;
     }
     sdkPlayer.WithPlayerAttribute((TCHAR_TO_UTF8(*attributeEntry.Key)), attribute);
   }
   sdkRequest.AddPlayer(sdkPlayer);
 }
 auto outcome = Aws::GameLift::Server::StartMatchBackfill(sdkRequest);
 if (outcome.IsSuccess()) {
   return FGameLiftStringOutcome(outcome.GetResult().GetTicketId());
 }
 else {
   return FGameLiftStringOutcome(FGameLiftError(outcome.GetError()));
 }
 #else
 return FGameLiftStringOutcome("");
 #endif
```

```
}
```

StopMatchBackfill()

Annulla una richiesta di match backfill attiva. Per ulteriori informazioni, consulta la funzione di FlexMatchriempimento.

Sintassi

```
FGameLiftGenericOutcome StopMatchBackfill (FStopMatchBackfillRequest
 &stopBackfillRequest);
```

Parametri

FStopMatchBackfillRequest

Un StopMatchBackfillRequest oggetto che identifica il ticket di matchmaking da annullare:

- L'ID del ticket assegnato alla richiesta di riempimento.
- Il matchmaker a cui è stata inviata la richiesta di riempimento.
- La sessione di gioco associata alla richiesta di riempimento.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
FGameLiftGenericOutcome FGameLiftServerSDKModule::StopMatchBackfill(const
 FStopMatchBackfillRequest& request)
{
  #if WITH_GAMELIFT
  Aws::GameLift::Server::Model::StopMatchBackfillRequest sdkRequest;
  sdkRequest.SetTicketId(TCHAR_TO_UTF8(*request.m_ticketId));
  sdkRequest.SetGameSessionArn(TCHAR_TO_UTF8(*request.m_gameSessionArn));

 sdkRequest.SetMatchmakingConfigurationArn(TCHAR_TO_UTF8(*request.m_matchmakingConfigurationArn
  auto outcome = Aws::GameLift::Server::StopMatchBackfill(sdkRequest);
  if (outcome.IsSuccess()) {
    return FGameLiftGenericOutcome(nullptr);
  }
  else {
```

```
    return FGameLiftGenericOutcome(FGameLiftError(outcome.GetError()));
  }
  #else
  return FGameLiftGenericOutcome(nullptr);
  #endif
}
```

GetComputeCertificate()

Recupera il percorso del certificato TLS utilizzato per crittografare la connessione di rete tra la risorsa di elaborazione Amazon GameLift Servers Anywhere e. Amazon GameLift Servers Puoi utilizzare il percorso del certificato quando registri il tuo dispositivo di elaborazione in una flotta Anywhere. Amazon GameLift Servers Per ulteriori informazioni, consultare RegisterCompute.

Sintassi

```
FGameLiftGetComputeCertificateOutcome FGameLiftServerSDKModule::GetComputeCertificate()
```

Valore restituito

Restituisce un `GetComputeCertificateResponse` oggetto contenente quanto segue:

- CertificatePath: il percorso del certificato TLS sulla risorsa di calcolo.
- HostName: il nome host della risorsa di elaborazione.

Esempio

```
FGameLiftGetComputeCertificateOutcome FGameLiftServerSDKModule::GetComputeCertificate()
{
  #if WITH_GAMELIFT
  auto outcome = Aws::GameLift::Server::GetComputeCertificate();
  if (outcome.IsSuccess()) {
    auto& outres = outcome.GetResult();
    FGameLiftGetComputeCertificateResult result;
    result.m_certificate_path = UTF8_TO_TCHAR(outres.GetCertificatePath());
    result.m_computeName = UTF8_TO_TCHAR(outres.GetComputeName());
    return FGameLiftGetComputeCertificateOutcome(result);
  }
  else {
    return FGameLiftGetComputeCertificateOutcome(FGameLiftError(outcome.GetError()));
  }
```

```
  #else
  return FGameLiftGetComputeCertificateOutcome(FGameLiftGetComputeCertificateResult());
  #endif
}
```

GetFleetRoleCredentials()

Recupera le credenziali del ruolo IAM che autorizzano Amazon GameLift Servers a interagire con altri. Servizi AWS Per ulteriori informazioni, consulta Comunica con altre AWS risorse delle tue flotte.

Sintassi

```
FGameLiftGetFleetRoleCredentialsOutcome
 FGameLiftServerSDKModule::GetFleetRoleCredentials(const
 FGameLiftGetFleetRoleCredentialsRequest &request)
```

Parametri

FGameLiftGetFleetRoleCredentialsRequest

Valore restituito

Restituisce un oggetto the section called “FGameLiftGetFleetRoleCredentialsOutcome”.

Esempio

```
FGameLiftGetFleetRoleCredentialsOutcome
 FGameLiftServerSDKModule::GetFleetRoleCredentials(const
 FGameLiftGetFleetRoleCredentialsRequest &request)
{
  #if WITH_GAMELIFT
  Aws::GameLift::Server::Model::GetFleetRoleCredentialsRequest sdkRequest;
  sdkRequest.SetRoleArn(TCHAR_TO_UTF8(*request.m_roleArn));
  sdkRequest.SetRoleSessionName(TCHAR_TO_UTF8(*request.m_roleSessionName));

  auto outcome = Aws::GameLift::Server::GetFleetRoleCredentials(sdkRequest);

  if (outcome.IsSuccess()) {
    auto& outres = outcome.GetResult();
    FGameLiftGetFleetRoleCredentialsResult result;
    result.m_assumedUserRoleArn = UTF8_TO_TCHAR(outres.GetAssumedUserRoleArn());
    result.m_assumedRoleId = UTF8_TO_TCHAR(outres.GetAssumedRoleId());
    result.m_accessKeyId = UTF8_TO_TCHAR(outres.GetAccessKeyId());
```

```
    result.m_secretAccessKey = UTF8_TO_TCHAR(outres.GetSecretAccessKey());
    result.m_sessionToken = UTF8_TO_TCHAR(outres.GetSessionToken());
    result.m_expiration = FDateTime::FromUnixTimestamp(outres.GetExpiration());
    return FGameLiftGetFleetRoleCredentialsOutcome(result);
  }
  else {
    return FGameLiftGetFleetRoleCredentialsOutcome(FGameLiftError(outcome.GetError()));
  }
  #else
  return
 FGameLiftGetFleetRoleCredentialsOutcome(FGameLiftGetFleetRoleCredentialsResult());
  #endif
}
```

Server SDK per Amazon GameLift Servers versione 4 e precedenti

Questo riferimento documenta l'SDK del server per Amazon GameLift Servers, versione 4.x e precedenti. L'SDK del server fornisce le funzionalità di base utilizzate dai server di gioco per comunicare con Amazon GameLift Servers servizio. Ad esempio, il server di gioco riceve dal servizio richieste di avvio e interruzione delle sessioni di gioco e fornisce aggiornamenti regolari sullo stato delle sessioni di gioco al servizio. Integra i tuoi server di gioco con l'SDK del server prima di distribuirli per l'hosting.

Usa questo riferimento all'SDK del server per integrare i tuoi server di gioco multiplayer personalizzati per l'hosting con Amazon GameLift Servers. Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

L'ultima versione principale del server SDK per Amazon GameLift Servers è 5.x. Le seguenti funzionalità di hosting richiedono aggiornamenti alla versione 5.x:

- Amazon GameLift Servers Ovunque
- Amazon GameLift Servers contenitori gestiti
- Amazon GameLift Servers plugin per Unreal Engine e Unity

Argomenti

- SDK per server C++ per Amazon GameLift Servers 4.x - Azioni
- SDK del server C# per Amazon GameLift Servers 4.x -- Azioni
- Server SDK (Unreal) per Amazon GameLift Servers -- Azioni

SDK per server C++ per Amazon GameLift Servers 4.x - Azioni

Usa il riferimento all'SDK del server per integrare il tuo gioco multiplayer con cui ospitarlo. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

> **Note**
>
> Questo riferimento si riferisce a una versione precedente del server SDK forAmazon GameLift Servers. Per la versione più recente, consulta Server C++ SDK 5.x per -- Azioni Amazon GameLift Servers.

SDK per server C++ per Amazon GameLift Servers 4.x -- Tipi di dati

Usa il riferimento all'SDK del server per integrare il tuo gioco multiplayer per l'hosting con Amazon GameLift Servers. Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

> **Note**
>
> Questo riferimento si riferisce a una versione precedente dell'SDK del server per Amazon GameLift Servers. Per la versione più recente, vediServer C++ SDK 5.x per Amazon GameLift Servers -- Tipi di dati.

L'API è definita in `GameLiftServerAPI.h`, `LogParameters.h` e `ProcessParameters.h`.

SDK per server C++ per Amazon GameLift Servers 4.x - Azioni

DescribePlayerSessionsRequest

Questo tipo di dati viene utilizzato per specificare quale sessione del giocatore recuperare. Puoi utilizzarlo come segue:

- Fornisci un PlayerSessionId messaggio per richiedere una sessione di gioco specifica.
- Fornisci un GameSessionId campo per richiedere tutte le sessioni dei giocatori nella sessione di gioco specificata.
- Fornisci un PlayerId campo per richiedere tutte le sessioni di gioco per il giocatore specificato.

Per le raccolte di grandi dimensioni delle sessioni giocatore, utilizza i parametri di paginazione per recuperare i risultati in blocchi sequenziali.

Indice

GameSessionId

Identificatore univoco della sessione di gioco. Utilizzare questo parametro per richiedere tutte le sessioni giocatore per la sessione di gioco specificata. Il formato dell'ID della sessione di gioco è il seguente: `arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>`. Il valore di **<**ID string**>** corrisponde a una stringa ID personalizzata o (se ne è stata specificata una al momento della creazione della sessione di gioco) a una stringa generata.

Tipo: string

Campo obbligatorio: no

Limite

Numero massimo di risultati da restituire. Utilizzate questo parametro con NextTokenper ottenere risultati sotto forma di set di pagine sequenziali. Se viene specificato un ID sessione giocatore, questo parametro verrà ignorato.

Tipo: integer

Campo obbligatorio: no

NextToken

Token che indica l'inizio della pagina sequenziale successiva relativa ai risultati. Utilizzare il token restituito con una chiamata precedente a questa operazione. Per specificare l'inizio del set di risultati, non specificare un valore. Se viene specificato un ID sessione giocatore, questo parametro verrà ignorato.

Tipo: string

Campo obbligatorio: no

PlayerId

Identificatore univoco per un giocatore. IDs I giocatori sono definiti dallo sviluppatore. Per informazioni, consulta Genera giocatore IDs.

Tipo: string

Campo obbligatorio: no

PlayerSessionId

Identificatore univoco della sessione giocatore.

Tipo: string

Campo obbligatorio: no

PlayerSessionStatusFilter

Stato sessione giocatore su cui filtrare i risultati. Tra gli stati sessione giocatore possibili sono inclusi i seguenti:

- RESERVED (RISERVATO) - La richiesta della sessione giocatore è stata ricevuta, ma il giocatore non si è ancora connesso al processo server e/o è stato convalidato.
- ACTIVE (ATTIVO) - Il giocatore è stato convalidato dal processo del server ed è attualmente collegato.
- COMPLETED (COMPLETATO) - La connessione del giocatore è stata interrotta.
- TIMEDOUT (SCADUTO) - È stata ricevuta la richiesta di una sessione giocatore, ma il giocatore non si è connesso e/o non è stato convalidato entro il limite di timeout (60 secondi).

Tipo: string

Campo obbligatorio: no

LogParameters

Questo tipo di dati viene utilizzato per identificare i file generati durante una sessione di gioco che desideri Amazon GameLift Servers da caricare e archiviare una volta terminata la sessione di gioco. Queste informazioni vengono comunicate al Amazon GameLift Servers servizio in una ProcessReady() chiamata.

Indice

logPaths

Percorsi di directory ai file di registro del server di gioco che desideri Amazon GameLift Servers da archiviare per accessi futuri. Questi file vengono generati durante ciascuna sessione di gioco. I percorsi e i nomi di file vengono definiti nel server di gioco e archiviati nella directory root della build di gioco. I percorsi di registro devono essere assoluti. Ad esempio, se la build di gioco

archivia i log della sessione di gioco in un percorso come `MyGame\sessionlogs\`, il percorso di log è `c:\game\MyGame\sessionLogs` (su un'istanza Windows) o `/local/game/MyGame/sessionLogs` (su un'istanza Linux).

Tipo: std:vector<std::string>

Campo obbligatorio: no

ProcessParameters

Questo tipo di dati contiene l'insieme di parametri inviati a Amazon GameLift Servers servizio in una ProcessReady() chiamata.

Indice

port

Numero di porta su cui il processo del server rimane in attesa di nuove connessioni dei giocatori. Il valore deve rientrare nella gamma di porte configurate per qualsiasi parco istanze che distribuisce questa build del server di gioco. Questo numero di porta è incluso nella sessione di gioco e negli oggetti della sessione del giocatore utilizzati dalle sessioni di gioco per la connessione a un processo server.

Tipo: integer

Campo obbligatorio: sì

logParameters

Un oggetto con un elenco di percorsi delle directory per i file log delle sessioni di gioco.

Tipo: Aws:GameLift: :Server:: LogParameters

Campo obbligatorio: no

onStartGameSessione

Nome della funzione di callback che Amazon GameLift Servers il servizio richiama per attivare una nuova sessione di gioco. Amazon GameLift Servers chiama questa funzione in risposta alla richiesta del client. CreateGameSession La funzione di callback passa un GameSessionoggetto (definito in Amazon GameLift Servers Riferimento all'API del servizio).

Tipo: `const std::function<void(Aws::GameLift::Model::GameSession)> onStartGameSession`

Campo obbligatorio: sì

onProcessTerminate

Nome della funzione di callback che Amazon GameLift Servers il servizio richiama per forzare la chiusura del processo del server. Dopo aver chiamato questa funzione, Amazon GameLift Servers attende cinque minuti che il processo del server si spenga e risponda con una ProcessEnding() chiamata. Se non viene ricevuta nessuna risposta, arresta il processo del server.

Tipo: `std::function<void()> onProcessTerminate`

Campo obbligatorio: no

onHealthCheck

Nome della funzione di callback che Amazon GameLift Servers il servizio richiama per richiedere un rapporto sullo stato di salute del processo del server. Amazon GameLift Servers chiama questa funzione ogni 60 secondi. Dopo aver chiamato questa funzione Amazon GameLift Servers attende 60 secondi per una risposta e, se non ne riceve nessuna. registra il processo del server come non integro.

Tipo: `std::function<bool()> onHealthCheck`

Campo obbligatorio: no

onUpdateGameSessione

Nome della funzione di callback che Amazon GameLift Servers service richiama per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift Servers chiama questa funzione quando è stata elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un GameSessionoggetto, uno status update (`updateReason`) e l'ID del ticket match backfill.

Tipo: `std::function<void(Aws::GameLift::Server::Model::UpdateGameSession)> onUpdateGameSession`

Campo obbligatorio: no

StartMatchBackfillRequest

Questo tipo di dati viene utilizzato per inviare una richiesta di backfill di abbinamento. Le informazioni vengono comunicate al Amazon GameLift Servers servizio in una StartMatchBackfill() chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco. L'operazione API GetGameSessionId() restituisce l'identificatore nel formato ARN.

Tipo: stringa

Campo obbligatorio: sì

MatchmakingConfigurationArn

Identificatore univoco, sotto forma di un ARN, che il matchmaker utilizza per questa richiesta. Per trovare il matchmaker utilizzato per creare la sessione di gioco originale, esaminare l'oggetto sessione di gioco nella proprietà dei dati del matchmaker. Scopri di più sui dati dei matchmaker in Word con i dati dei matchmaker.

Tipo: stringa

Campo obbligatorio: sì

Players

Un set di dati che rappresenta tutti i giocatori che sono attualmente impegnati nella sessione di gioco. Il matchmaker utilizza queste informazioni per cercare nuovi giocatori che rappresentano un buon abbinamento per i giocatori attuali. Vedi il Amazon GameLift Servers Guida di riferimento API per una descrizione del formato degli oggetti Player. Per trovare gli attributi dei giocatori e gli incarichi della squadra, cerca nell'oggetto della sessione di gioco, nella proprietà dei dati del matchmaker. IDs Se la latenza viene utilizzata dal matchmaker, raccogliere la latenza aggiornata per la regione attuale e includerla nei dati di ciascun giocatore.

Tipo: std:vectorhttps://docs.aws.amazon.com/gamelift/latest/apireference/API_Player.html <player>

Campo obbligatorio: sì

TicketId

Identificatore univoco per un abbinamento o un ticket di richiesta di backfill degli abbinamenti. Se non viene fornito alcun valore qui, Amazon GameLift Servers ne genererà uno sotto forma di UUID. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill degli abbinamenti o annullare la richiesta, se necessario.

Tipo: string

Campo obbligatorio: no

StopMatchBackfillRequest

Questo tipo di dati viene utilizzato per annullare una richiesta di backfill di abbinamento. Le informazioni vengono comunicate al Amazon GameLift Servers servizio in una StopMatchBackfill() chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco associato alla richiesta in fase di annullamento.

Tipo: stringa

Campo obbligatorio: sì

MatchmakingConfigurationArn

Identificatore univoco del matchmaker a cui è stata inviata questa richiesta.

Tipo: stringa

Campo obbligatorio: sì

TicketId

Identificatore univoco del ticket di richiesta di backfill da annullare.

Tipo: stringa

Campo obbligatorio: sì

SDK per server C++ per Amazon GameLift Servers 4.x -- Tipi di dati

Argomenti

- AcceptPlayerSession()
- ActivateGameSession()
- DescribePlayerSessions()

- GetGameSessionId()
- GetInstanceCertificate()
- GetSdkVersion()
- GetTerminationTime()
- InitSDK()
- ProcessEnding()
- ProcessReady()
- ProcessReadyAsync()
- RemovePlayerSession()
- StartMatchBackfill()
- StopMatchBackfill()
- TerminateGameSession()
- UpdatePlayerSessionCreationPolicy()
- Distruggi ()

AcceptPlayerSession()

Notifica al Amazon GameLift Servers servizio che un giocatore con l'ID di sessione del giocatore specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift Serversverifica che l'ID della sessione del giocatore sia valido, ovvero che l'ID giocatore abbia riservato uno slot nella sessione di gioco. Dopo la convalida, Amazon GameLift Servers modifica lo stato dello slot giocatore da RESERVED (RISERVATO) ad ACTIVE (ATTIVO).

Sintassi

```
GenericOutcome AcceptPlayerSession(const std::string& playerSessionId);
```

Parametri

playerSessionId

ID univoco rilasciato dal Amazon GameLift Servers servizio in risposta a una chiamata all'azione dell'API AWS SDKAmazon GameLift Servers. CreatePlayerSession Il client di gioco fa riferimento a questo ID durante la connessione al processo del server.

Tipo: std: :string

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra una funzione per la gestione di una richiesta di connessione, inclusa la convalida e il rifiuto di una sessione giocatore non valida. IDs

```
void ReceiveConnectingPlayerSessionID (Connection& connection, const std::string&
 playerSessionId){
    Aws::GameLift::GenericOutcome connectOutcome =
        Aws::GameLift::Server::AcceptPlayerSession(playerSessionId);
    if(connectOutcome.IsSuccess())
    {
        connectionToSessionMap.emplace(connection, playerSessionId);
        connection.Accept();
    }
    else
    {
        connection.Reject(connectOutcome.GetError().GetMessage();
    }
}
```

ActivateGameSession()

Notifica al servizio Amazon GameLift Servers che il processo del server ha avviato una sessione di gioco ed è pronto per ricevere le connessioni dei giocatori. Questa operazione deve essere chiamata come parte della funzione di callback `onStartGameSession()`, dopo il completamento dell'inizializzazione di tutte le sessioni di gioco.

Sintassi

```
GenericOutcome ActivateGameSession();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra la chiamata a `ActivateGameSession()` nell'ambito della funzione di callback on`StartGameSession()`.

```
void onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
   // game-specific tasks when starting a new game session, such as loading map
   GenericOutcome outcome = Aws::GameLift::Server::ActivateGameSession();
}
```

DescribePlayerSessions()

Recupera i dati della sessione giocatore, tra cui le impostazioni, i metadati della sessione e i dati dei giocatori. Utilizza questa operazione per ottenere le informazioni per una singola sessione giocatore, per tutte le sessioni giocatore in una sessione di gioco o per tutte le sessioni giocatore associate a un singolo ID giocatore.

Sintassi

```
DescribePlayerSessionsOutcome DescribePlayerSessions (
    const Aws::GameLift::Server::Model::DescribePlayerSessionsRequest
 &describePlayerSessionsRequest);
```

Parametri

describePlayerSessionsRichiesta

Un oggetto DescribePlayerSessionsRequest che descrive le sessioni giocatore da recuperare.

Campo obbligatorio: sì

Valore restituito

Se l'esito è positivo, restituisce un oggetto `DescribePlayerSessionsOutcome` contenente un set di oggetti di sessione giocatore corrispondente ai parametri della richiesta. Gli oggetti della sessione Player hanno una struttura identica al tipo di PlayerSessiondati dell'Amazon GameLift ServersAPI AWS SDK.

Esempio

Questo esempio illustra una richiesta per tutte le sessioni giocatore attivamente connesse a una sessione di gioco specificata. Se ometti `NextToken` e imposti il valore `Limit` su 10, Amazon GameLift Servers restituisce i primi 10 record di sessioni giocatore che corrispondono alla richiesta.

```
// Set request parameters
Aws::GameLift::Server::Model::DescribePlayerSessionsRequest request;
request.SetPlayerSessionStatusFilter(Aws::GameLift::Server::Model::PlayerSessionStatusMapper::G
request.SetLimit(10);
request.SetGameSessionId("the game session ID");    // can use GetGameSessionId()

// Call DescribePlayerSessions
Aws::GameLift::DescribePlayerSessionsOutcome playerSessionsOutcome =
    Aws::GameLift::Server::DescribePlayerSessions(request);
```

GetGameSessionId()

Recupera un identificatore univoco della sessione di gioco attualmente ospitata dal processo del server, se il processo del server è attivo. L'identificatore viene restituito nel formato ARN: `arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>`.

Per i processi inattivi che non sono ancora stati attivati con una sessione di gioco, la chiamata restituisce `Success` = `True` e `GameSessionId` = `""` (una stringa vuota).

Sintassi

```
AwsStringOutcome GetGameSessionId();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Se l'esito è positivo, l'ID della sessione di gioco verrà restituito come oggetto `AwsStringOutcome`. Se l'esito è negativo, verrà restituito un messaggio di errore.

Esempio

```
Aws::GameLift::AwsStringOutcome sessionIdOutcome =
    Aws::GameLift::Server::GetGameSessionId();
```

GetInstanceCertificate()

Recupera la posizione del file di un certificato TLS con codifica pem associato alla flotta e alle relative istanze. AWS Certificate Manager genera questo certificato quando crei un nuovo parco veicoli con la configurazione del certificato impostata su GENERATED. Utilizza questo certificato per stabilire una connessione sicura con un client di gioco e per crittografare la comunicazione client/server.

Sintassi

```
GetInstanceCertificateOutcome GetInstanceCertificate();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce un `GetInstanceCertificateOutcome` oggetto contenente la posizione del file di certificato TLS e della catena di certificati della flotta, che sono archiviati nell'istanza. Nell'istanza viene inoltre archiviato un file di certificato radice, estratto dalla catena di certificati. Se l'esito è negativo, verrà restituito un messaggio di errore.

Per ulteriori informazioni sul certificato e sui dati della catena di certificati, consulta GetCertificate Response Elements nell' AWS Certificate Manager API Reference.

Esempio

```
Aws::GameLift::GetInstanceCertificateOutcome certificateOutcome =
    Aws::GameLift::Server::GetInstanceCertificate();
```

GetSdkVersion()

Restituisce il numero di versione corrente dell'SDK in uso.

Sintassi

```
AwsStringOutcome GetSdkVersion();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Se l'esito è positivo, restituisce la versione corrente dell'SDK come oggetto `AwsStringOutcome`. La stringa restituita include solo il numero di versione (ad esempio «3.1.5"). Se l'esito è negativo, verrà restituito un messaggio di errore.

Esempio

```
Aws::GameLift::AwsStringOutcome SdkVersionOutcome =
    Aws::GameLift::Server::GetSdkVersion();
```

GetTerminationTime()

Restituisce il tempo di arresto pianificato di un processo del server, se è disponibile un tempo di chiusura. Un processo server esegue questa azione dopo aver ricevuto una `onProcessTerminate()` richiamata dal servizio. Amazon GameLift Servers Amazon GameLift Serverspuò effettuare una chiamata `onProcessTerminate()` per i seguenti motivi: (1) quando il processo server ha segnalato problemi di salute o non ha rispostoAmazon GameLift Servers, (2) quando chiude l'istanza durante un evento di scale-down o (3) quando un'istanza viene interrotta a causa di un'interruzione Spot.

Se il processo ha ricevuto un `onProcessTerminate()` callback, il valore restituito è il tempo di terminazione stimato. Se il processo non ha ricevuto una `onProcessTerminate()` richiamata, viene restituito un messaggio di errore. Ulteriori informazioni sull'arresto di un processo del server.

Sintassi

```
AwsLongOutcome GetTerminationTime();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce l'ora di terminazione come oggetto`AwsLongOutcome`. Il valore è l'ora di terminazione, espressa in segni di spunta trascorsi dalle 00:01 00:00:00. Ad esempio, il valore data/ora 2020-09-13 12:26:40 -000Z è uguale a 637355968000000000 ticks. Se non è disponibile alcun orario di terminazione, restituisce un messaggio di errore.

Esempio

```
Aws::GameLift::AwsLongOutcome TermTimeOutcome =
    Aws::GameLift::Server::GetTerminationTime();
```

InitSDK()

Inizializza l'SDK Amazon GameLift Servers. Questo metodo deve essere chiamato all'avvio, prima che avvenga qualsiasi altra inizializzazione correlata a Amazon GameLift Servers.

Sintassi

```
InitSDKOutcome InitSDK();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce un InitSdkOutcome oggetto che indica che il processo del server è pronto per la chiamataProcessReady().

Esempio

```
Aws::GameLift::Server::InitSDKOutcome initOutcome =
    Aws::GameLift::Server::InitSDK();
```

ProcessEnding()

Avvisa il servizio Amazon GameLift Servers che il processo del server è in fase di arresto. Questo metodo deve essere richiamato dopo tutte le altre attività di pulizia, tra cui l'arresto di tutte le sessioni di gioco attive. Questo metodo deve uscire con il codice 0; un codice di uscita diverso da zero genera un messaggio di evento che indica che il processo non è terminato correttamente.

Una volta terminato il metodo con un codice 0, è possibile terminare il processo con un codice di uscita corretto. È inoltre possibile uscire dal processo con un codice di errore. Se si esce con un codice di errore, l'evento fleet indicherà che il processo è terminato in modo anomalo ()SERVER_PROCESS_TERMINATED_UNHEALTHY.

Sintassi

```
GenericOutcome ProcessEnding();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
Aws::GameLift::GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
if (outcome.Success)
    exit(0);  // exit with success
// otherwise, exit with error code
exit(errorCode);
```

ProcessReady()

Notifica al servizio Amazon GameLift Servers che il processo del server è pronto per l'hosting delle sessioni di gioco. Chiama questo metodo dopo aver richiamato InitSDK() e completato con successo le attività di configurazione necessarie prima che il processo del server possa ospitare una sessione di gioco. Questo metodo deve essere chiamato solo una volta per processo.

Questa chiamata è sincrona. Per effettuare una chiamata asincrona, utilizza ProcessReadyAsync(). Per ulteriori dettagli, consulta Inizializza il processo del server.

Sintassi

```
GenericOutcome ProcessReady(
    const Aws::GameLift::Server::ProcessParameters &processParameters);
```

Parametri

processParameters

Un oggetto ProcessParameters che comunica le informazioni seguenti sul processo del server:

- Nomi di metodi di callback, implementati nel codice del server di gioco, che il servizio Amazon GameLift Servers richiama per comunicare con il processo del server.
- Numero di porta sulla quale è in ascolto il processo del server.
- Percorso verso qualsiasi file specifico di una sessione di gioco che Amazon GameLift Servers deve acquisire e archiviare.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra le implementazioni della funzione di chiamata e callback ProcessReady().

```
// Set parameters and call ProcessReady
std::string serverLog("serverOut.log");        // Example of a log file written by the
 game server
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);

int listenPort = 9339;

Aws::GameLift::Server::ProcessParameters processReadyParameter =
 Aws::GameLift::Server::ProcessParameters(
    std::bind(&Server::onStartGameSession, this, std::placeholders::_1),
    std::bind(&Server::onProcessTerminate, this),
    std::bind(&Server::OnHealthCheck, this),
    std::bind(&Server::OnUpdateGameSession, this),
    listenPort,
    Aws::GameLift::Server::LogParameters(logPaths));

Aws::GameLift::GenericOutcome outcome =
   Aws::GameLift::Server::ProcessReady(processReadyParameter);

// Implement callback functions
void Server::onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
   // game-specific tasks when starting a new game session, such as loading map
   GenericOutcome outcome =
       Aws::GameLift::Server::ActivateGameSession (maxPlayers);
```

```
}

void Server::onProcessTerminate()
{
   // game-specific tasks required to gracefully shut down a game session,
   // such as notifying players, preserving game state data, and other cleanup
   GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
}

bool Server::onHealthCheck()
{
    bool health;
    // complete health evaluation within 60 seconds and set health
    return health;
}
```

ProcessReadyAsync()

Notifica al servizio Amazon GameLift Servers che il processo del server è pronto per l'hosting delle sessioni di gioco. Questo metodo deve essere chiamato quando il processo del server è pronto per l'hosting di una sessione di gioco. I parametri specificano i nomi delle funzioni di callback che Amazon GameLift Servers deve chiamare in determinate condizioni. Il codice del server di gioco deve implementare queste funzioni.

Questa chiamata è asincrona. Per effettuare una chiamata sincrona, utilizza ProcessReady(). Per ulteriori dettagli, consulta Inizializza il processo del server.

Sintassi

```
GenericOutcomeCallable ProcessReadyAsync(
    const Aws::GameLift::Server::ProcessParameters &processParameters);
```

Parametri

processParameters

Un oggetto ProcessParameters che comunica le informazioni seguenti sul processo del server:

- Nomi di metodi di callback, implementati nel codice del server di gioco, che il servizio Amazon GameLift Servers richiama per comunicare con il processo del server.
- Numero di porta sulla quale è in ascolto il processo del server.

- Percorso verso qualsiasi file specifico di una sessione di gioco che Amazon GameLift Servers deve acquisire e archiviare.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da esito positivo o negativo con un messaggio di errore.

Esempio

```
// Set parameters and call ProcessReady
std::string serverLog("serverOut.log");        // This is an example of a log file
 written by the game server
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);

int listenPort = 9339;

Aws::GameLift::Server::ProcessParameters processReadyParameter =
 Aws::GameLift::Server::ProcessParameters(
    std::bind(&Server::onStartGameSession, this, std::placeholders::_1),
    std::bind(&Server::onProcessTerminate, this),
    std::bind(&Server::OnHealthCheck, this),
    std::bind(&Server::OnUpdateGameSession, this),
    listenPort,
    Aws::GameLift::Server::LogParameters(logPaths));

Aws::GameLift::GenericOutcomeCallable outcome =
   Aws::GameLift::Server::ProcessReadyAsync(processReadyParameter);

// Implement callback functions
void onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
   // game-specific tasks when starting a new game session, such as loading map
   GenericOutcome outcome = Aws::GameLift::Server::ActivateGameSession (maxPlayers);
}

void onProcessTerminate()
{
   // game-specific tasks required to gracefully shut down a game session,
   // such as notifying players, preserving game state data, and other cleanup
   GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
```

```
}

bool onHealthCheck()
{
    // perform health evaluation and complete within 60 seconds
    return health;
}
```

RemovePlayerSession()

Notifica al servizio Amazon GameLift Servers che un giocatore con l'ID della sessione giocatore specificato si è disconnesso dal processo del server. In risposta, Amazon GameLift Servers modifica lo slot giocatore su disponibile, consentendone l'assegnazione a un nuovo giocatore.

Sintassi

```
GenericOutcome RemovePlayerSession(
    const std::string& playerSessionId);
```

Parametri

playerSessionId

ID univoco rilasciato dal Amazon GameLift Servers servizio in risposta a una chiamata all'azione CreatePlayerSessiondell'Amazon GameLift ServersAPI AWS SDK. Il client di gioco fa riferimento a questo ID durante la connessione al processo del server.

Tipo: std: :string

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

```
Aws::GameLift::GenericOutcome disconnectOutcome =
    Aws::GameLift::Server::RemovePlayerSession(playerSessionId);
```

StartMatchBackfill()

Invia una richiesta per trovare nuovi giocatori per gli slot aperti in una sessione di gioco creata con FlexMatch. Vedi anche l'azione AWS SDK StartMatchBackfill(). Con questa operazione, è possibile avviare le richieste di backfill degli abbinamenti da un processo del server di gioco che ospita la sessione di gioco. Scopri di più sulla funzione di FlexMatchriempimento.

Questa operazione è asincrona. Se i nuovi giocatori vengono abbinati con successo, il Amazon GameLift Servers servizio fornisce dati aggiornati del matchmaker richiamando la funzione di callback. `OnUpdateGameSession()`

Un processo del server può avere un solo backfill degli abbinamenti attivo alla volta. Per inviare una nuova richiesta, chiama prima StopMatchBackfill() per annullare la richiesta originale.

Sintassi

```
StartMatchBackfillOutcome StartMatchBackfill (
    const Aws::GameLift::Server::Model::StartMatchBackfillRequest
 &startBackfillRequest);
```

Parametri

StartMatchBackfillRequest

Un oggetto StartMatchBackfillRequest che comunica le informazioni seguenti:

- ID del ticket da assegnare alla richiesta di backfill. Questa informazione è facoltativa; se l'ID non viene fornito, Amazon GameLift Servers ne creerà automaticamente uno.
- Matchmaker a cui inviare la richiesta. L'ARN di configurazione completo è obbligatorio. Questo valore può essere acquisito dai dati del matchmaker della sessione di gioco.
- ID della sessione di gioco che è in fase di backfilling.
- Dati di abbinamento disponibili per i giocatori correnti della sessione di gioco.

Campo obbligatorio: sì

Valore restituito

Restituisce un StartMatchBackfillOutcome oggetto con il match backfill ticket o un errore con un messaggio di errore. Lo stato del ticket può essere monitorato utilizzando l'azione AWS DescribeMatchmaking SDK ().

Esempio

```
// Build a backfill request
std::vector<Player> players;
Aws::GameLift::Server::Model::StartMatchBackfillRequest startBackfillRequest;
startBackfillRequest.SetTicketId("a ticket ID");
  //optional, autogenerated if not provided
startBackfillRequest.SetMatchmakingConfigurationArn("the matchmaker configuration
 ARN"); //from the game session matchmaker data
startBackfillRequest.SetGameSessionArn("the game session ARN");
  // can use GetGameSessionId()
startBackfillRequest.SetPlayers(players);
    //from the game session matchmaker data

// Send backfill request
Aws::GameLift::StartMatchBackfillOutcome backfillOutcome =
    Aws::GameLift::Server::StartMatchBackfill(startBackfillRequest);

// Implement callback function for backfill
void Server::OnUpdateGameSession(Aws::GameLift::Server::Model::GameSession gameSession,
 Aws::GameLift::Server::Model::UpdateReason updateReason, std::string backfillTicketId)
{
   // handle status messages
   // perform game-specific tasks to prep for newly matched players
}
```

StopMatchBackfill()

Annulla una richiesta di backfill degli abbinamenti attiva creata con StartMatchBackfill(). Vedi anche l'azione AWS StopMatchmaking SDK (). Scopri di più sulla funzione di FlexMatchriempimento.

Sintassi

```
GenericOutcome StopMatchBackfill (
    const Aws::GameLift::Server::Model::StopMatchBackfillRequest &stopBackfillRequest);
```

Parametri

StopMatchBackfillRequest

Un oggetto StopMatchBackfillRequest che identifica il ticket di abbinamento da annullare:

- ID del ticket assegnato alla richiesta di backfill in fase di annullamento

- matchmaker a cui è stata inviata la richiesta di backfill
- sessione di gioco associata alla richiesta di backfill

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
// Set backfill stop request parameters

Aws::GameLift::Server::Model::StopMatchBackfillRequest stopBackfillRequest;
stopBackfillRequest.SetTicketId("the ticket ID");
stopBackfillRequest.SetGameSessionArn("the game session ARN");
  // can use GetGameSessionId()
stopBackfillRequest.SetMatchmakingConfigurationArn("the matchmaker configuration ARN");
  // from the game session matchmaker data

Aws::GameLift::GenericOutcome stopBackfillOutcome =
    Aws::GameLift::Server::StopMatchBackfillRequest(stopBackfillRequest);
```

TerminateGameSession()

Questo metodo è obsoleto con la versione 4.0.1. Invece, il processo del server dovrebbe richiamare ProcessEnding() dopo la fine di una sessione di gioco.

Notifica al Amazon GameLift Servers servizio che il processo del server ha terminato la sessione di gioco corrente. Questa azione viene richiamata quando il processo del server rimane attivo e pronto per ospitare una nuova sessione di gioco. Dovrebbe essere chiamata solo dopo il completamento della procedura di interruzione della sessione di gioco, perché segnala Amazon GameLift Servers che il processo del server è immediatamente disponibile per ospitare una nuova sessione di gioco.

Questa azione non viene eseguita se il processo del server verrà chiuso dopo l'interruzione della sessione di gioco. Invece, chiama ProcessEnding() per segnalare che sia la sessione di gioco che il processo sul server stanno terminando.

Sintassi

```
GenericOutcome TerminateGameSession();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

UpdatePlayerSessionCreationPolicy()

Aggiorna la capacità della sessione di gioco corrente di accettare nuove sessioni giocatore. Una sessione di gioco può essere configurata per accettare o rifiutare tutte le nuove sessioni giocatore. Vedi anche l'azione AWS SDK UpdateGameSession().

Sintassi

```
GenericOutcome UpdatePlayerSessionCreationPolicy(
    Aws::GameLift::Model::PlayerSessionCreationPolicy newPlayerSessionPolicy);
```

Parametri

newPlayerSessionPolitica

Valore della stringa che indica se la sessione di gioco accetta nuovi giocatori.

Tipo: Aws:GameLift: :Model:: PlayerSessionCreationPolicy enum. I valori validi includono:

- ACCEPT_ALL: accetta tutte le nuove sessioni giocatore.
- DENY_ALL: rifiuta tutte le nuove sessioni giocatore.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio definisce la policy di partecipazione alla sessione di gioco corrente per accettare tutti i giocatori.

```
Aws::GameLift::GenericOutcome outcome =
 Aws::GameLift::Server::UpdatePlayerSessionCreationPolicy(Aws::GameLift::Model::PlayerSessionCr
```

Distruggi ()

Pulisce la memoria allocata da initSDK () durante l'inizializzazione del server di gioco. Utilizza questo metodo dopo aver terminato un processo del server di gioco per evitare di sprecare memoria del server.

Sintassi

```
GenericOutcome Aws::GameLift::Server::Destroy();
```

Parametri

Non ci sono parametri.

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio pulisce la memoria allocata da InitSDK al termine di un processo sul server di gioco.

```
if (Aws::GameLift::Server::ProcessEnding().IsSuccess()) {
  Aws::GameLift::Server::Destroy();
  exit(0);
}
```

SDK del server C# per Amazon GameLift Servers 4.x -- Azioni

Usa il riferimento all'SDK del server per integrare il tuo gioco multiplayer con cui ospitarlo. Amazon GameLift Servers Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

> **Note**
>
> Questo riferimento si riferisce a una versione precedente del server SDK forAmazon GameLift Servers. Per la versione più recente, consulta Server C# SDK 5.x per -- Azioni Amazon GameLift Servers.

SDK per server C# per Amazon GameLift Servers 4.x -- Tipi di dati

Usa il riferimento all'SDK del server per integrare il tuo gioco multiplayer per l'hosting con Amazon GameLift Servers. Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

> **Note**
>
> Questo riferimento si riferisce a una versione precedente dell'SDK del server per Amazon GameLift Servers. Per la versione più recente, vediServer C# SDK 5.x per Amazon GameLift Servers -- Tipi di dati.

SDK del server C# per Amazon GameLift Servers 4.x -- Azioni

Argomenti

- LogParameters
- DescribePlayerSessionsRequest
- ProcessParameters
- StopMatchBackfillRequest

LogParameters

Questo tipo di dati viene utilizzato per identificare i file generati durante una sessione di gioco che desideri Amazon GameLift Servers da caricare e archiviare una volta terminata la sessione di gioco. Queste informazioni vengono comunicate al Amazon GameLift Servers servizio in una ProcessReady() chiamata.

Indice

logPaths

Elenco dei percorsi di directory dei file di registro del server di gioco che desideri Amazon GameLift Servers da archiviare per accessi futuri. Questi file vengono generati da un processo server durante ogni sessione di gioco; i percorsi dei file e i nomi sono definiti nel server di gioco e memorizzati nella directory root build di gioco. I percorsi di registro devono essere assoluti. Ad esempio, se la build di gioco archivia i log della sessione di gioco in un percorso come `MyGame`

\sessionlogs\, il percorso di log è c:\game\MyGame\sessionLogs (su un'istanza Windows) o /local/game/MyGame/sessionLogs (su un'istanza Linux).

Tipo: List<String>

Campo obbligatorio: no

DescribePlayerSessionsRequest

Questo tipo di dati viene utilizzato per specificare quale sessione del giocatore recuperare. Può essere utilizzato in diversi modi: (1) fornire PlayerSessionId a per richiedere una sessione di gioco specifica; (2) fornire GameSessionId a per richiedere tutte le sessioni dei giocatori nella sessione di gioco specificata; o (3) fornire PlayerId a per richiedere tutte le sessioni di gioco per il giocatore specificato. Per le raccolte di grandi dimensioni delle sessioni giocatore, utilizzare i parametri di paginazione per recuperare i risultati in pagine sequenziali.

Indice

GameSessionId

Identificatore univoco della sessione di gioco. Utilizzare questo parametro per richiedere tutte le sessioni giocatore per la sessione di gioco specificata. Il formato dell'ID della sessione di gioco è il seguente: arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>. Il valore di <ID string> corrisponde a una stringa ID personalizzata o (se ne è stata specificata una al momento della creazione della sessione di gioco) a una stringa generata.

Tipo: string

Campo obbligatorio: no

Limite

Numero massimo di risultati da restituire. Utilizzate questo parametro con NextTokenper ottenere risultati sotto forma di set di pagine sequenziali. Se viene specificato un ID sessione giocatore, questo parametro verrà ignorato.

Tipo: integer

Campo obbligatorio: no

NextToken

Token che indica l'inizio della pagina sequenziale successiva relativa ai risultati. Utilizzare il token restituito con una chiamata precedente a questa operazione. Per specificare l'inizio del set di risultati, non specificare un valore. Se viene specificato un ID sessione giocatore, questo parametro verrà ignorato.

Tipo: string

Campo obbligatorio: no

PlayerId

Identificatore univoco per un giocatore. IDs I giocatori sono definiti dallo sviluppatore. Per informazioni, consulta Genera giocatore IDs.

Tipo: string

Campo obbligatorio: no

PlayerSessionId

Identificatore univoco della sessione giocatore.

Tipo: string

Campo obbligatorio: no

PlayerSessionStatusFilter

Stato sessione giocatore su cui filtrare i risultati. Tra gli stati sessione giocatore possibili sono inclusi i seguenti:

- RESERVED (RISERVATO) - La richiesta della sessione giocatore è stata ricevuta, ma il giocatore non si è ancora connesso al processo server e/o è stato convalidato.
- ACTIVE (ATTIVO) - Il giocatore è stato convalidato dal processo del server ed è attualmente collegato.
- COMPLETED (COMPLETATO) - La connessione del giocatore è stata interrotta.
- TIMEDOUT (SCADUTO) - È stata ricevuta la richiesta di una sessione giocatore, ma il giocatore non si è connesso e/o non è stato convalidato entro il limite di timeout (60 secondi).

Tipo: string

Campo obbligatorio: no

ProcessParameters

Questo tipo di dati contiene l'insieme di parametri inviati al Amazon GameLift Servers servizio in una ProcessReady() chiamata.

Indice

port

Numero di porta a cui il processo del server rimarrà in attesa di nuove connessioni dei giocatori. Il valore deve rientrare nella gamma di porte configurate per qualsiasi parco istanze che distribuisce questa build del server di gioco. Questo numero di porta è incluso nella sessione di gioco e negli oggetti della sessione del giocatore utilizzati dalle sessioni di gioco per la connessione a un processo server.

Tipo: integer

Campo obbligatorio: sì

logParameters

Un oggetto con un elenco di percorsi delle directory per i file log delle sessioni di gioco.

Tipo: `Aws::GameLift::Server::LogParameters`

Campo obbligatorio: sì

onStartGameSessione

Nome della funzione di callback che Amazon GameLift Servers il servizio richiama per attivare una nuova sessione di gioco. Amazon GameLift Servers chiama questa funzione in risposta alla richiesta del client. CreateGameSession La funzione di callback accetta un GameSessionoggetto (definito in Amazon GameLift Servers Riferimento all'API del servizio).

Tipo: `void OnStartGameSessionDelegate(GameSession gameSession)`

Campo obbligatorio: sì

onProcessTerminate

Nome della funzione di callback che Amazon GameLift Servers il servizio richiama per forzare la chiusura del processo del server. Dopo aver chiamato questa funzione, Amazon GameLift Servers attende cinque minuti che il processo server si chiuda e risponda con una ProcessEnding() chiamata prima di chiudere il processo server.

Tipo: `void OnProcessTerminateDelegate()`

Campo obbligatorio: sì

onHealthCheck

Nome della funzione di callback che Amazon GameLift Servers il servizio richiama per richiedere un rapporto sullo stato di salute del processo del server. Amazon GameLift Servers chiama questa funzione ogni 60 secondi. Dopo aver chiamato questa funzione Amazon GameLift Servers attende 60 secondi per una risposta e, se non ne riceve nessuna. registra il processo del server come non integro.

Tipo: `bool OnHealthCheckDelegate()`

Campo obbligatorio: sì

onUpdateGameSessione

Nome della funzione di callback che Amazon GameLift Servers service richiama per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift Servers chiama questa funzione quando è stata elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un GameSessionoggetto, uno status update (`updateReason`) e l'ID del ticket match backfill.

Tipo: `void OnUpdateGameSessionDelegate ( UpdateGameSession updateGameSession )`

Campo obbligatorio: no

StartMatchBackfillRequest

Questo tipo di dati viene utilizzato per inviare una richiesta di backfill di abbinamento. Le informazioni vengono comunicate al Amazon GameLift Servers servizio in una StartMatchBackfill() chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco. Il metodo SDK GetGameSessionId() restituisce l'identificatore in formato ARN.

Tipo: stringa

Campo obbligatorio: sì

MatchmakingConfigurationArn

Identificatore univoco, sotto forma di un ARN, che il matchmaker utilizza per questa richiesta. Per trovare il matchmaker utilizzato per creare la sessione di gioco originale, esaminare l'oggetto sessione di gioco nella proprietà dei dati del matchmaker. Scopri di più sui dati dei matchmaker in Work with matchmaker data.

Tipo: stringa

Campo obbligatorio: sì

Players

Un set di dati che rappresenta tutti i giocatori che sono attualmente impegnati nella sessione di gioco. Il matchmaker utilizza queste informazioni per cercare nuovi giocatori che rappresentano un buon abbinamento per i giocatori attuali. Vedi il Amazon GameLift Servers Guida di riferimento API per una descrizione del formato degli oggetti Player. Per trovare gli attributi dei giocatori e gli incarichi della squadra, cerca nell'oggetto della sessione di gioco, nella proprietà dei dati del matchmaker. IDs Se la latenza viene utilizzata dal matchmaker, raccogliere la latenza aggiornata per la regione attuale e includerla nei dati di ciascun giocatore.

Tipo: Giocatore[ ]

Campo obbligatorio: sì

TicketId

Identificatore univoco per un abbinamento o un ticket di richiesta di backfill degli abbinamenti. Se non viene fornito alcun valore qui, Amazon GameLift Servers ne genererà uno sotto forma di UUID. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill degli abbinamenti o annullare la richiesta, se necessario.

Tipo: string

Campo obbligatorio: no

StopMatchBackfillRequest

Questo tipo di dati viene utilizzato per annullare una richiesta di backfill di abbinamento. Le informazioni vengono comunicate al Amazon GameLift Servers servizio in una StopMatchBackfill() chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco associato alla richiesta in fase di annullamento.

Tipo: stringa

Campo obbligatorio: sì

MatchmakingConfigurationArn

Identificatore univoco del matchmaker a cui è stata inviata questa richiesta.

Tipo: stringa

Campo obbligatorio: sì

TicketId

Identificatore univoco del ticket di richiesta di backfill da annullare.

Tipo: stringa

Campo obbligatorio: sì

SDK per server C# per Amazon GameLift Servers 4.x -- Tipi di dati

Argomenti

- AcceptPlayerSession()
- ActivateGameSession()
- DescribePlayerSessions()
- GetGameSessionId()
- GetInstanceCertificate()
- GetSdkVersion()
- GetTerminationTime()
- InitSDK()
- ProcessEnding()

- ProcessReady()
- RemovePlayerSession()
- StartMatchBackfill()
- StopMatchBackfill()
- TerminateGameSession()
- UpdatePlayerSessionCreationPolicy()

AcceptPlayerSession()

Notifica al Amazon GameLift Servers servizio che un giocatore con l'ID di sessione del giocatore specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift Serversverifica che l'ID della sessione del giocatore sia valido, ovvero che l'ID giocatore abbia riservato uno slot nella sessione di gioco. Dopo la convalida, Amazon GameLift Servers modifica lo stato dello slot giocatore da RESERVED (RISERVATO) ad ACTIVE (ATTIVO).

Sintassi

```
GenericOutcome AcceptPlayerSession(String playerSessionId)
```

Parametri

playerSessionId

ID univoco rilasciato Amazon GameLift Servers quando viene creata una nuova sessione giocatore. Un ID di sessione del giocatore viene specificato in un `PlayerSession` oggetto, che viene restituito in risposta a una chiamata del client alle azioni GameLift API StartGameSessionPlacement CreateGameSession, DescribeGameSessionPlacement, o DescribePlayerSessions.

Tipo: stringa

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra una funzione per la gestione di una richiesta di connessione, inclusa la convalida e il rifiuto di una sessione giocatore non valida. IDs

```
void ReceiveConnectingPlayerSessionID (Connection connection, String playerSessionId){
    var acceptPlayerSessionOutcome =
 GameLiftServerAPI.AcceptPlayerSession(playerSessionId);
     if(acceptPlayerSessionOutcome.Success)
    {
        connectionToSessionMap.emplace(connection, playerSessionId);
        connection.Accept();
    }
     else
    {
        connection.Reject(acceptPlayerSessionOutcome.Error.ErrorMessage);    }
}
```

ActivateGameSession()

Notifica al servizio Amazon GameLift Servers che il processo del server ha attivato una sessione di gioco ed è pronto per ricevere le connessioni dei giocatori. Questa operazione deve essere chiamata come parte della funzione di callback `onStartGameSession()`, dopo il completamento dell'inizializzazione di tutte le sessioni di gioco.

Sintassi

```
GenericOutcome ActivateGameSession()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra la chiamata a `ActivateGameSession()` nell'ambito della funzione delegata `onStartGameSession()`.

```
void OnStartGameSession(GameSession gameSession)
{
    // game-specific tasks when starting a new game session, such as loading map

    // When ready to receive players
    var activateGameSessionOutcome = GameLiftServerAPI.ActivateGameSession();
}
```

DescribePlayerSessions()

Recupera i dati della sessione giocatore, tra cui le impostazioni, i metadati della sessione e i dati dei giocatori. Utilizza questa operazione per ottenere le informazioni per una singola sessione giocatore, per tutte le sessioni giocatore in una sessione di gioco o per tutte le sessioni giocatore associate a un singolo ID giocatore.

Sintassi

```
DescribePlayerSessionsOutcome DescribePlayerSessions(DescribePlayerSessionsRequest
 describePlayerSessionsRequest)
```

Parametri

describePlayerSessionsRichiesta

Un oggetto DescribePlayerSessionsRequest che descrive le sessioni giocatore da recuperare.

Campo obbligatorio: sì

Valore restituito

Se l'esito è positivo, restituisce un oggetto `DescribePlayerSessionsOutcome` contenente un set di oggetti di sessione giocatore corrispondente ai parametri della richiesta. Gli oggetti della sessione Player hanno una struttura identica al tipo di PlayerSessiondati dell'Amazon GameLift ServersAPI AWS SDK.

Esempio

Questo esempio illustra una richiesta per tutte le sessioni giocatore attivamente connesse a una sessione di gioco specificata. Omettendo NextTokene impostando il valore Limite su 10, Amazon

GameLift Servers verranno restituiti i record delle prime 10 sessioni di gioco corrispondenti alla richiesta.

```
// Set request parameters
var describePlayerSessionsRequest = new
 Aws.GameLift.Server.Model.DescribePlayerSessionsRequest()
{
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result,    //gets the ID for
 the current game session
    Limit = 10,
    PlayerSessionStatusFilter =
 PlayerSessionStatusMapper.GetNameForPlayerSessionStatus(PlayerSessionStatus.ACTIVE)
};
// Call DescribePlayerSessions
Aws::GameLift::DescribePlayerSessionsOutcome playerSessionsOutcome =
    Aws::GameLift::Server::Model::DescribePlayerSessions(describePlayerSessionRequest);
```

GetGameSessionId()

Recupera l'ID della sessione di gioco attualmente ospitata dal processo del server, se il processo del server è attivo.

Per i processi inattivi che non sono ancora stati attivati con una sessione di gioco, la chiamata restituisce `Success` = `True` e `GameSessionId` = `""` (una stringa vuota).

Sintassi

```
AwsStringOutcome GetGameSessionId()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Se l'esito è positivo, l'ID della sessione di gioco verrà restituito come oggetto `AwsStringOutcome`. Se l'esito è negativo, verrà restituito un messaggio di errore.

Esempio

```
var getGameSessionIdOutcome = GameLiftServerAPI.GetGameSessionId();
```

GetInstanceCertificate()

Recupera la posizione del file di un certificato TLS con codifica pem associato alla flotta e alle relative istanze. AWS Certificate Manager genera questo certificato quando crei un nuovo parco veicoli con la configurazione del certificato impostata su GENERATED. Utilizza questo certificato per stabilire una connessione sicura con un client di gioco e per crittografare la comunicazione client/server.

Sintassi

```
GetInstanceCertificateOutcome GetInstanceCertificate();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce un `GetInstanceCertificateOutcome` oggetto contenente la posizione del file di certificato TLS e della catena di certificati della flotta, che sono archiviati nell'istanza. Nell'istanza viene inoltre archiviato un file di certificato radice, estratto dalla catena di certificati. Se l'esito è negativo, verrà restituito un messaggio di errore.

Per ulteriori informazioni sul certificato e sui dati della catena di certificati, consulta GetCertificate Response Elements nell' AWS Certificate Manager API Reference.

Esempio

```
var getInstanceCertificateOutcome = GameLiftServerAPI.GetInstanceCertificate();
```

GetSdkVersion()

Restituisce il numero di versione corrente dell'SDK integrato nel processo del server.

Sintassi

```
AwsStringOutcome GetSdkVersion()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Se l'esito è positivo, restituisce la versione corrente dell'SDK come oggetto `AwsStringOutcome`. La stringa restituita include solo il numero di versione (ad esempio «3.1.5"). Se l'esito è negativo, verrà restituito un messaggio di errore.

Esempio

```
var getSdkVersionOutcome = GameLiftServerAPI.GetSdkVersion();
```

GetTerminationTime()

Restituisce il tempo di arresto pianificato di un processo del server, se è disponibile un tempo di chiusura. Un processo server esegue questa azione dopo aver ricevuto una `onProcessTerminate()` richiamata dal servizio. Amazon GameLift Servers Amazon GameLift Serverspuò effettuare una chiamata `onProcessTerminate()` per i seguenti motivi: (1) per problemi di salute (il processo del server ha segnalato lo stato della porta o non ha rispostoAmazon GameLift Servers, (2) quando termina l'istanza durante un evento di scale-down o (3) quando un'istanza viene terminata a causa di un'interruzione dell'istanza spot.

Se il processo ha ricevuto una `onProcessTerminate()` richiamata, il valore restituito è il tempo di terminazione stimato. Se il processo non ha ricevuto una `onProcessTerminate()` richiamata, viene restituito un messaggio di errore. Ulteriori informazioni sull'arresto di un processo del server.

Sintassi

```
AwsDateTimeOutcome GetTerminationTime()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce l'ora di terminazione come oggetto`AwsDateTimeOutcome`. Il valore è l'ora di terminazione, espressa in segni di spunta trascorsi dalle 00:01 00:00:00. Ad esempio, il valore data/ora 2020-09-13 12:26:40 -000Z è uguale a 637355968000000000 ticks. Se non è disponibile alcun orario di terminazione, restituisce un messaggio di errore.

Esempio

```
var getTerminationTimeOutcome = GameLiftServerAPI.GetTerminationTime();
```

InitSDK()

Inizializza l'SDK Amazon GameLift Servers. Questo metodo deve essere chiamato all'avvio, prima che avvenga qualsiasi altra inizializzazione correlata a Amazon GameLift Servers.

Sintassi

```
InitSDKOutcome InitSDK()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce un InitSdkOutcome oggetto che indica che il processo del server è pronto per la chiamataProcessReady().

Esempio

```
var initSDKOutcome = GameLiftServerAPI.InitSDK();
```

ProcessEnding()

Avvisa il servizio Amazon GameLift Servers che il processo del server è in fase di arresto. Questo metodo deve essere richiamato dopo tutte le altre attività di pulizia, tra cui l'arresto di tutte le sessioni di gioco attive. Questo metodo deve uscire con il codice 0; un codice di uscita diverso da zero genera un messaggio di evento che indica che il processo non è terminato correttamente.

Una volta terminato il metodo con un codice 0, è possibile terminare il processo con un codice di uscita corretto. È inoltre possibile uscire dal processo con un codice di errore. Se si esce con un codice di errore, l'evento fleet indicherà che il processo è terminato in modo anomalo ()SERVER_PROCESS_TERMINATED_UNHEALTHY.

Sintassi

```
GenericOutcome ProcessEnding()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
var processEndingOutcome = GameLiftServerAPI.ProcessEnding();
if (processReadyOutcome.Success)
   Environment.Exit(0);
// otherwise, exit with error code
Environment.Exit(errorCode);
```

ProcessReady()

Notifica al servizio Amazon GameLift Servers che il processo del server è pronto per l'hosting delle sessioni di gioco. Chiama questo metodo dopo aver richiamato InitSDK() e completato con successo le attività di configurazione necessarie prima che il processo del server possa ospitare una sessione di gioco. Questo metodo deve essere chiamato solo una volta per processo.

Sintassi

```
GenericOutcome ProcessReady(ProcessParameters processParameters)
```

Parametri

processParameters

Un oggetto ProcessParameters che comunica le informazioni seguenti sul processo del server:

- Nomi di metodi di callback, implementati nel codice del server di gioco, che il servizio Amazon GameLift Servers richiama per comunicare con il processo del server.
- Numero di porta sulla quale è in ascolto il processo del server.
- Percorso verso qualsiasi file specifico di una sessione di gioco che Amazon GameLift Servers deve acquisire e archiviare.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra la chiamata ProcessReady() e le implementazioni della funzione delegata.

```
// Set parameters and call ProcessReady
var processParams = new ProcessParameters(
   this.OnGameSession,
   this.OnProcessTerminate,
   this.OnHealthCheck,
   this.OnGameSessionUpdate,
   port,
   new LogParameters(new List<string>()          // Examples of log and error files
 written by the game server
   {
      "C:\\game\\logs",
      "C:\\game\\error"
   })
);

var processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);

// Implement callback functions
void OnGameSession(GameSession gameSession)
{
   // game-specific tasks when starting a new game session, such as loading map
   // When ready to receive players
   var activateGameSessionOutcome = GameLiftServerAPI.ActivateGameSession();
}

void OnProcessTerminate()
{
   // game-specific tasks required to gracefully shut down a game session,
   // such as notifying players, preserving game state data, and other cleanup
    var ProcessEndingOutcome = GameLiftServerAPI.ProcessEnding();
}

bool OnHealthCheck()
{
    bool isHealthy;
    // complete health evaluation within 60 seconds and set health
```

```
    return isHealthy;
}
```

RemovePlayerSession()

Notifica al servizio Amazon GameLift Servers che un giocatore con l'ID della sessione giocatore specificato si è disconnesso dal processo del server. In risposta, Amazon GameLift Servers modifica lo slot giocatore su disponibile, consentendone l'assegnazione a un nuovo giocatore.

Sintassi

```
GenericOutcome RemovePlayerSession(String playerSessionId)
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon GameLift Servers quando viene creata una nuova sessione di gioco. Un ID di sessione del giocatore viene specificato in un `PlayerSession` oggetto, che viene restituito in risposta a una chiamata del client alle azioni GameLift API StartGameSessionPlacement CreateGameSession, DescribeGameSessionPlacement, o DescribePlayerSessions.

Tipo: stringa

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
Aws::GameLift::GenericOutcome disconnectOutcome =
    Aws::GameLift::Server::RemovePlayerSession(playerSessionId);
```

StartMatchBackfill()

Invia una richiesta per trovare nuovi giocatori per gli slot aperti in una sessione di gioco creata con FlexMatch. Vedi anche l'azione AWS SDK StartMatchBackfill(). Con questa operazione, è possibile

avviare le richieste di backfill degli abbinamenti da un processo del server di gioco che ospita la sessione di gioco. Scopri di più sulla funzione di FlexMatchriempimento.

Questa operazione è asincrona. Se i nuovi giocatori sono stati abbinati in modo corretto, il servizio Amazon GameLift Servers invia i dati del matchmaker aggiornati tramite la funzione di callback `OnUpdateGameSession()`.

Un processo del server può avere un solo backfill degli abbinamenti attivo alla volta. Per inviare una nuova richiesta, chiama prima StopMatchBackfill() per annullare la richiesta originale.

Sintassi

```
StartMatchBackfillOutcome StartMatchBackfill (StartMatchBackfillRequest
 startBackfillRequest);
```

Parametri

StartMatchBackfillRequest

Un oggetto StartMatchBackfillRequest che comunica le informazioni seguenti:

- ID del ticket da assegnare alla richiesta di backfill. Questa informazione è facoltativa; se l'ID non viene fornito, Amazon GameLift Servers ne creerà automaticamente uno.
- Matchmaker a cui inviare la richiesta. L'ARN di configurazione completo è obbligatorio. Questo valore può essere acquisito dai dati del matchmaker della sessione di gioco.
- ID della sessione di gioco che è in fase di backfilling.
- Dati di abbinamento disponibili per i giocatori correnti della sessione di gioco.

Campo obbligatorio: sì

Valore restituito

Restituisce un StartMatchBackfillOutcome oggetto con l'ID del ticket Match Backfill o restituisce un errore con un messaggio di errore.

Esempio

```
// Build a backfill request
var startBackfillRequest = new AWS.GameLift.Server.Model.StartMatchBackfillRequest()
{
    TicketId = "a ticket ID", //optional
```

```
    MatchmakingConfigurationArn = "the matchmaker configuration ARN",
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result,    // gets ID for
 current game session
        //get player data for all currently connected players
            MatchmakerData matchmakerData =
              MatchmakerData.FromJson(gameSession.MatchmakerData);  // gets matchmaker
 data for current players
            // get matchmakerData.Players
            // remove data for players who are no longer connected
    Players = ListOfPlayersRemainingInTheGame
};

// Send backfill request
var startBackfillOutcome = GameLiftServerAPI.StartMatchBackfill(startBackfillRequest);

// Implement callback function for backfill
void OnUpdateGameSession(GameSession myGameSession)
{
   // game-specific tasks to prepare for the newly matched players and update
 matchmaker data as needed
}
```

StopMatchBackfill()

Annulla una richiesta di backfill degli abbinamenti attiva creata con StartMatchBackfill(). Vedi anche l'azione AWS SDK StopMatchmaking(). Scopri di più sulla funzione di FlexMatchriempimento.

Sintassi

```
GenericOutcome StopMatchBackfill (StopMatchBackfillRequest stopBackfillRequest);
```

Parametri

StopMatchBackfillRequest

Un oggetto StopMatchBackfillRequest che identifica il ticket di abbinamento da annullare:

- ID del ticket assegnato alla richiesta di backfill in fase di annullamento
- matchmaker a cui è stata inviata la richiesta di backfill
- sessione di gioco associata alla richiesta di backfill

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

```
// Set backfill stop request parameters

var stopBackfillRequest = new AWS.GameLift.Server.Model.StopMatchBackfillRequest()
{
    TicketId = "a ticket ID", //optional, if not provided one is autogenerated
    MatchmakingConfigurationArn = "the matchmaker configuration ARN", //from the game
 session matchmaker data
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result    //gets the ID for
 the current game session
};

var stopBackfillOutcome =
 GameLiftServerAPI.StopMatchBackfillRequest(stopBackfillRequest);
```

TerminateGameSession()

Questo metodo è obsoleto con la versione 4.0.1. Invece, il processo del server dovrebbe richiamare ProcessEnding() dopo la fine di una sessione di gioco.

Notifica al Amazon GameLift Servers servizio che il processo del server ha terminato la sessione di gioco corrente. Questa azione viene richiamata quando il processo del server rimane attivo e pronto per ospitare una nuova sessione di gioco. Dovrebbe essere chiamata solo dopo il completamento della procedura di interruzione della sessione di gioco, perché segnala Amazon GameLift Servers che il processo del server è immediatamente disponibile per ospitare una nuova sessione di gioco.

Questa azione non viene eseguita se il processo del server verrà chiuso dopo l'interruzione della sessione di gioco. Invece, chiama ProcessEnding() per segnalare che sia la sessione di gioco che il processo sul server stanno terminando.

Sintassi

```
GenericOutcome TerminateGameSession()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra un processo del server al termine di una sessione di gioco.

```
// game-specific tasks required to gracefully shut down a game session,
// such as notifying players, preserving game state data, and other cleanup

var terminateGameSessionOutcome = GameLiftServerAPI.TerminateGameSession();
var processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);
```

UpdatePlayerSessionCreationPolicy()

Aggiorna la capacità della sessione di gioco corrente di accettare nuove sessioni giocatore. Una sessione di gioco può essere configurata per accettare o rifiutare tutte le nuove sessioni giocatore. (Vedi anche l'azione UpdateGameSession() nel Amazon GameLift ServersService API Reference).

Sintassi

```
GenericOutcome UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy
 playerSessionPolicy)
```

Parametri

newPlayerSessionPolitica

Valore della stringa che indica se la sessione di gioco accetta nuovi giocatori.

Tipo: enum. PlayerSessionCreationPolicy. I valori validi includono:

- ACCEPT_ALL: accetta tutte le nuove sessioni giocatore.
- DENY_ALL: rifiuta tutte le nuove sessioni giocatore.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio definisce la policy di partecipazione alla sessione di gioco corrente per accettare tutti i giocatori.

```
var updatePlayerSessionCreationPolicyOutcomex =

 GameLiftServerAPI.UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy.ACCEPT_ALL);
```

Server SDK (Unreal) per Amazon GameLift Servers -- Azioni

Usa l'SDK del server per Unreal per integrare il tuo gioco multiplayer per l'hosting con Amazon GameLift Servers. Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

> **Note**
>
> Questo riferimento si riferisce a una versione precedente dell'SDK del server per Amazon GameLift Servers. Per la versione più recente, vediServer C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers.

Questa API è definita in `GameLiftServerSDK.h` e `GameLiftServerSDKModels.h`.

Per configurare il plugin Unreal Engine e vedere esempi di codice Integrazione Amazon GameLift Servers in un progetto Unreal Engine.

Server SDK (Unreal) per Amazon GameLift Servers -- Tipi di dati

Usa il Amazon GameLift Servers server SDK for Unreal reference con cui integrare il gioco multiplayer per l'hosting con Amazon GameLift Servers. Per indicazioni sul processo di integrazione, consultaAggiungi Amazon GameLift Servers al tuo server di gioco.

> **Note**
>
> Questo riferimento si riferisce a una versione precedente del server SDK per Amazon GameLift Servers. Per la versione più recente, vediServer C++ (Unreal) SDK 5.x per Amazon GameLift Servers -- Tipi di dati.

Questa API è definita in `GameLiftServerSDK.h` e `GameLiftServerSDKModels.h`.

Per configurare il plugin Unreal Engine e vedere esempi di codice Integrazione Amazon GameLift Servers in un progetto Unreal Engine.

Server SDK (Unreal) per Amazon GameLift Servers -- Azioni

Argomenti

- FDescribePlayerSessionsRequest
- FProcessParametri
- FStartMatchBackfillRequest
- FStopMatchBackfillRequest

FDescribePlayerSessionsRequest

Questo tipo di dati viene utilizzato per specificare quale sessione del giocatore recuperare. Puoi utilizzarlo come segue:

- Fornisci un PlayerSessionId messaggio per richiedere una sessione di gioco specifica.
- Fornisci un GameSessionId campo per richiedere tutte le sessioni dei giocatori nella sessione di gioco specificata.
- Fornisci un PlayerId campo per richiedere tutte le sessioni di gioco per il giocatore specificato.

Per le raccolte di grandi dimensioni delle sessioni giocatore, utilizza i parametri di paginazione per recuperare i risultati in blocchi sequenziali.

Indice

GameSessionId

Identificatore univoco della sessione di gioco. Utilizzare questo parametro per richiedere tutte le sessioni giocatore per la sessione di gioco specificata. Il formato dell'ID della sessione di gioco è il seguente: `arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>`. Il valore di <ID string> corrisponde a una stringa ID personalizzata o (se ne è stata specificata una al momento della creazione della sessione di gioco) a una stringa generata.

Tipo: string

Campo obbligatorio: no

Limite

Numero massimo di risultati da restituire. Utilizzate questo parametro con NextTokenper ottenere risultati sotto forma di set di pagine sequenziali. Se viene specificato un ID sessione giocatore, questo parametro verrà ignorato.

Tipo: integer

Campo obbligatorio: no

NextToken

Token che indica l'inizio della pagina sequenziale successiva relativa ai risultati. Utilizzare il token restituito con una chiamata precedente a questa operazione. Per specificare l'inizio del set di risultati, non specificare un valore. Se viene specificato un ID sessione giocatore, questo parametro verrà ignorato.

Tipo: string

Campo obbligatorio: no

PlayerId

Identificatore univoco per un giocatore. IDs I giocatori sono definiti dallo sviluppatore. Per informazioni, consulta Genera giocatore IDs.

Tipo: string

Campo obbligatorio: no

PlayerSessionId

Identificatore univoco della sessione giocatore.

Tipo: string

Campo obbligatorio: no

PlayerSessionStatusFilter

Stato sessione giocatore su cui filtrare i risultati. Tra gli stati sessione giocatore possibili sono inclusi i seguenti:

- RESERVED (RISERVATO) - La richiesta della sessione giocatore è stata ricevuta, ma il giocatore non si è ancora connesso al processo server e/o è stato convalidato.

- ACTIVE (ATTIVO) - Il giocatore è stato convalidato dal processo del server ed è attualmente collegato.
- COMPLETED (COMPLETATO) - La connessione del giocatore è stata interrotta.
- TIMEDOUT (SCADUTO) - È stata ricevuta la richiesta di una sessione giocatore, ma il giocatore non si è connesso e/o non è stato convalidato entro il limite di timeout (60 secondi).

Tipo: string

Campo obbligatorio: no

FProcessParametri

Questo tipo di dati contiene l'insieme di parametri inviati a Amazon GameLift Servers servizio in una ProcessReady() chiamata.

Indice

port

Numero di porta a cui il processo del server rimarrà in attesa di nuove connessioni dei giocatori. Il valore deve rientrare nella gamma di porte configurate per qualsiasi parco istanze che distribuisce questa build del server di gioco. Questo numero di porta è incluso nella sessione di gioco e negli oggetti della sessione del giocatore utilizzati dalle sessioni di gioco per la connessione a un processo server.

Tipo: integer

Campo obbligatorio: sì

logParameters

Un oggetto con un elenco di percorsi delle directory per i file log delle sessioni di gioco.

Tipo: TArray < FString >

Campo obbligatorio: no

onStartGameSessione

Nome della funzione di callback che Amazon GameLift Servers il servizio richiama per attivare una nuova sessione di gioco. Amazon GameLift Servers chiama questa funzione in risposta alla richiesta del client. CreateGameSession La funzione di callback accetta un GameSessionoggetto (definito in Amazon GameLift Servers Riferimento all'API del servizio).

Tipo: FOnStartGameSession

Campo obbligatorio: sì

onProcessTerminate

Nome della funzione di callback che il Amazon GameLift Servers il servizio richiama per forzare la chiusura del processo del server. Dopo aver chiamato questa funzione, Amazon GameLift Servers attende cinque minuti che il processo server si chiuda e risponda con una ProcessEnding() chiamata prima di chiudere il processo server.

Tipo: delegato FSimple

Campo obbligatorio: no

onHealthCheck

Nome della funzione di callback che Amazon GameLift Servers il servizio richiama per richiedere un rapporto sullo stato di salute del processo del server. Amazon GameLift Servers chiama questa funzione ogni 60 secondi. Dopo aver chiamato questa funzione Amazon GameLift Servers attende 60 secondi per una risposta e, se non ne riceve nessuna. registra il processo del server come non integro.

Tipo: FOnHealthCheck

Campo obbligatorio: no

onUpdateGameSessione

Nome della funzione di callback che il Amazon GameLift Servers service richiama per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift Servers chiama questa funzione quando è stata elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un GameSessionoggetto, uno status update (`updateReason`) e l'ID del ticket match backfill.

Tipo: FOnUpdateGameSession

Campo obbligatorio: no

FStartMatchBackfillRequest

Questo tipo di dati viene utilizzato per inviare una richiesta di backfill di abbinamento. Le informazioni vengono comunicate al Amazon GameLift Servers servizio in una StartMatchBackfill() chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco. L'operazione API GetGameSessionId() restituisce l'identificatore nel formato ARN.

Tipo: FString

Campo obbligatorio: sì

MatchmakingConfigurationArn

Identificatore univoco, sotto forma di un ARN, che il matchmaker utilizza per questa richiesta. Per trovare il matchmaker utilizzato per creare la sessione di gioco originale, esaminare l'oggetto sessione di gioco nella proprietà dei dati del matchmaker. Scopri di più sui dati dei matchmaker in Lavora con i dati dei matchmaker.

Tipo: FString

Campo obbligatorio: sì

Players

Un set di dati che rappresenta tutti i giocatori che sono attualmente impegnati nella sessione di gioco. Il matchmaker utilizza queste informazioni per cercare nuovi giocatori che rappresentano un buon abbinamento per i giocatori attuali. Vedi il Amazon GameLift Servers Guida di riferimento API per una descrizione del formato degli oggetti Player. Per trovare gli attributi dei giocatori e gli incarichi della squadra, cerca nell'oggetto della sessione di gioco, nella proprietà dei dati del matchmaker. IDs Se la latenza viene utilizzata dal matchmaker, raccogliere la latenza aggiornata per la regione attuale e includerla nei dati di ciascun giocatore.

Digita: < > TArray FPlayer

Campo obbligatorio: sì

TicketId

Identificatore univoco per un abbinamento o un ticket di richiesta di backfill degli abbinamenti. Se non viene fornito alcun valore qui, Amazon GameLift Servers ne genererà uno sotto forma di UUID. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill degli abbinamenti o annullare la richiesta, se necessario.

Tipo: FString

Campo obbligatorio: no

FStopMatchBackfillRequest

Questo tipo di dati viene utilizzato per annullare una richiesta di backfill di abbinamento. Le informazioni vengono comunicate al Amazon GameLift Servers servizio in una StopMatchBackfill() chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco associato alla richiesta in fase di annullamento.

Tipo: FString

Campo obbligatorio: sì

MatchmakingConfigurationArn

Identificatore univoco del matchmaker a cui è stata inviata questa richiesta.

Tipo: FString

Campo obbligatorio: sì

TicketId

Identificatore univoco del ticket di richiesta di backfill da annullare.

Tipo: FString

Campo obbligatorio: sì

Server SDK (Unreal) per Amazon GameLift Servers -- Tipi di dati

Argomenti

- AcceptPlayerSession()
- ActivateGameSession()

- DescribePlayerSessions()
- GetGameSessionId()
- GetInstanceCertificate()
- GetSdkVersion()
- InitSDK()
- ProcessEnding()
- ProcessReady()
- RemovePlayerSession()
- StartMatchBackfill()
- StopMatchBackfill()
- TerminateGameSession()
- UpdatePlayerSessionCreationPolicy()

AcceptPlayerSession()

Notifica il Amazon GameLift Servers servizio che un giocatore con l'ID di sessione del giocatore specificato si è connesso al processo del server e necessita di convalida. Amazon GameLift Servers verifica che l'ID della sessione del giocatore sia valido, ovvero che l'ID giocatore abbia riservato uno slot nella sessione di gioco. Una volta convalidato, Amazon GameLift Servers modifica lo stato dello slot del giocatore da RISERVATO ad ATTIVO.

Sintassi

```
FGameLiftGenericOutcome AcceptPlayerSession(const FString& playerSessionId)
```

Parametri

playerSessionId

ID univoco emesso dal Amazon GameLift Servers servizio in risposta a una chiamata all' AWS SDK Amazon GameLift Servers azione API. CreatePlayerSession Il client di gioco fa riferimento a questo ID durante la connessione al processo del server.

Tipo: FString

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

ActivateGameSession()

Notifica il Amazon GameLift Servers servizio che il processo del server ha attivato una sessione di gioco ed è ora pronto a ricevere le connessioni dei giocatori. Questa operazione deve essere chiamata come parte della funzione di callback `onStartGameSession()`, dopo il completamento dell'inizializzazione di tutte le sessioni di gioco.

Sintassi

```
FGameLiftGenericOutcome ActivateGameSession()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

DescribePlayerSessions()

Recupera i dati della sessione giocatore, tra cui le impostazioni, i metadati della sessione e i dati dei giocatori. Utilizza questa operazione per ottenere le informazioni per una singola sessione giocatore, per tutte le sessioni giocatore in una sessione di gioco o per tutte le sessioni giocatore associate a un singolo ID giocatore.

Sintassi

```
FGameLiftDescribePlayerSessionsOutcome DescribePlayerSessions(const
 FGameLiftDescribePlayerSessionsRequest &describePlayerSessionsRequest)
```

Parametri

describePlayerSessionsRichiesta

Un oggetto FDescribePlayerSessionsRequest che descrive le sessioni giocatore da recuperare.

Campo obbligatorio: sì

Valore restituito

Se l'esito è positivo, restituisce un oggetto FDescribePlayerSessionsRequest contenente un set di oggetti di sessione giocatore corrispondente ai parametri della richiesta. Gli oggetti della sessione Player hanno una struttura identica a quella dell' AWS SDK Amazon GameLift Servers Tipo di PlayerSessiondati API.

GetGameSessionId()

Recupera l'ID della sessione di gioco attualmente ospitata dal processo del server, se il processo del server è attivo.

Sintassi

```
FGameLiftStringOutcome GetGameSessionId()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Se l'esito è positivo, l'ID della sessione di gioco verrà restituito come oggetto `FGameLiftStringOutcome`. Se l'esito è negativo, verrà restituito un messaggio di errore.

GetInstanceCertificate()

Recupera la posizione del file di un certificato TLS con codifica pem associato alla flotta e alle relative istanze. AWS Certificate Manager genera questo certificato quando crei un nuovo parco veicoli con la configurazione del certificato impostata su GENERATED. Utilizza questo certificato per stabilire una connessione sicura con un client di gioco e per crittografare la comunicazione client/server.

Sintassi

```
FGameLiftGetInstanceCertificateOutcome GetInstanceCertificate()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce un `GetInstanceCertificateOutcome` oggetto contenente la posizione del file di certificato TLS e della catena di certificati della flotta, che sono archiviati nell'istanza. Nell'istanza viene inoltre archiviato un file di certificato radice, estratto dalla catena di certificati. Se l'esito è negativo, verrà restituito un messaggio di errore.

Per ulteriori informazioni sul certificato e sui dati della catena di certificati, consulta GetCertificate Response Elements nell' AWS Certificate Manager API Reference.

GetSdkVersion()

Restituisce il numero di versione corrente dell'SDK integrato nel processo del server.

Sintassi

```
FGameLiftStringOutcome GetSdkVersion();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Se l'esito è positivo, restituisce la versione corrente dell'SDK come oggetto `FGameLiftStringOutcome`. La stringa restituita include solo il numero di versione (ad esempio «3.1.5"). Se l'esito è negativo, verrà restituito un messaggio di errore.

Esempio

```
Aws::GameLift::AwsStringOutcome SdkVersionOutcome =
    Aws::GameLift::Server::GetSdkVersion();
```

InitSDK()

Inizializza il Amazon GameLift Servers SDK. Questo metodo dovrebbe essere chiamato all'avvio, prima di ogni altro Amazon GameLift Serverssi verifica l'inizializzazione correlata.

Sintassi

```
FGameLiftGenericOutcome InitSDK()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico costituito da esito positivo o negativo con un messaggio di errore.

ProcessEnding()

Notifica il Amazon GameLift Servers servizio che il processo del server si sta chiudendo. Questo metodo deve essere richiamato dopo tutte le altre attività di pulizia, tra cui l'arresto di tutte le sessioni di gioco attive. Questo metodo deve uscire con il codice 0; un codice di uscita diverso da zero genera un messaggio di evento che indica che il processo non è terminato correttamente.

Sintassi

```
FGameLiftGenericOutcome ProcessEnding()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

ProcessReady()

Notifica il Amazon GameLift Servers servizio che il processo del server è pronto per ospitare sessioni di gioco. Richiama questo metodo dopo aver richiamato InitSDK() e completato con successo le attività di configurazione necessarie prima che il processo del server possa ospitare una sessione di gioco. Questo metodo deve essere chiamato solo una volta per processo.

Sintassi

```
FGameLiftGenericOutcome ProcessReady(FProcessParameters &processParameters)
```

Parametri

FProcessParametri

Un oggetto FProcessParametri che comunica le informazioni seguenti sul processo del server:

- Nomi dei metodi di callback, implementati nel codice del server di gioco, che Amazon GameLift Servers il servizio richiama per comunicare con il processo del server.
- Numero di porta sulla quale è in ascolto il processo del server.
- Percorso a qualsiasi file specifico della sessione di gioco che desideri Amazon GameLift Servers da acquisire e archiviare.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

Esempio

Consulta il codice di esempio nella sezione relativa all'utilizzo del plug-in Unreal Engine.

RemovePlayerSession()

Notifica il Amazon GameLift Servers servizio che un giocatore con l'ID di sessione del giocatore specificato si è disconnesso dal processo del server. In risposta, Amazon GameLift Servers modifica lo slot del giocatore rendendolo disponibile, il che consente di assegnarlo a un nuovo giocatore.

Sintassi

```
FGameLiftGenericOutcome RemovePlayerSession(const FString& playerSessionId)
```

Parametri

playerSessionId

ID univoco emesso dal Amazon GameLift Servers servizio in risposta a una chiamata all' AWS SDK Amazon GameLift Servers azione API. CreatePlayerSession Il client di gioco fa riferimento a questo ID durante la connessione al processo del server.

Tipo: FString

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

StartMatchBackfill()

Invia una richiesta per trovare nuovi giocatori per gli slot aperti in una sessione di gioco creata con FlexMatch. Vedi anche l'azione AWS SDK StartMatchBackfill(). Con questa operazione, è possibile avviare le richieste di backfill degli abbinamenti da un processo del server di gioco che ospita la sessione di gioco. Ulteriori informazioni sull'FlexMatch funzione di riempimento.

Questa operazione è asincrona. Se i nuovi giocatori vengono abbinati con successo, il Amazon GameLift Servers il servizio fornisce dati aggiornati sul matchmaker utilizzando la funzione di callback. `OnUpdateGameSession()`

Un processo del server può avere un solo backfill degli abbinamenti attivo alla volta. Per inviare una nuova richiesta, chiama prima StopMatchBackfill() per annullare la richiesta originale.

Sintassi

```
FGameLiftStringOutcome StartMatchBackfill (FStartMatchBackfillRequest
 &startBackfillRequest);
```

Parametri

FStartMatchBackfillRequest

Un oggetto FStartMatchBackfillRequest che comunica le informazioni seguenti:

- ID del ticket da assegnare alla richiesta di backfill. Queste informazioni sono facoltative; se non viene fornito alcun ID, Amazon GameLift Servers ne genererà uno automaticamente.
- Matchmaker a cui inviare la richiesta. L'ARN di configurazione completo è obbligatorio. Questo valore può essere acquisito dai dati del matchmaker della sessione di gioco.
- ID della sessione di gioco che è in fase di backfilling.
- Dati di abbinamento disponibili per i giocatori correnti della sessione di gioco.

Campo obbligatorio: sì

Valore restituito

Se completato, restituisce il ticket di backfill degli abbinamenti come oggetto `FGameLiftStringOutcome`. Se l'esito è negativo, verrà restituito un messaggio di errore. Lo stato del ticket può essere monitorato utilizzando l'azione AWS SDK (). DescribeMatchmaking

StopMatchBackfill()

Annulla una richiesta di backfill degli abbinamenti attiva creata con StartMatchBackfill(). Vedi anche l'azione AWS StopMatchmaking SDK (). Ulteriori informazioni sull'FlexMatch funzione di riempimento.

Sintassi

```
FGameLiftGenericOutcome StopMatchBackfill (FStopMatchBackfillRequest
 &stopBackfillRequest);
```

Parametri

StopMatchBackfillRequest

Un oggetto FStopMatchBackfillRequest che identifica il ticket di abbinamento da annullare:

- ID del ticket assegnato alla richiesta di backfill in fase di annullamento
- matchmaker a cui è stata inviata la richiesta di backfill
- sessione di gioco associata alla richiesta di backfill

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico costituito da successo o fallimento con un messaggio di errore.

TerminateGameSession()

Questo metodo è obsoleto con la versione 4.0.1. Invece, il processo del server dovrebbe richiamare ProcessEnding() dopo la fine di una sessione di gioco.

Notifica il Amazon GameLift Servers servizio che il processo del server ha terminato la sessione di gioco corrente. Questa azione viene richiamata quando il processo del server rimane attivo e pronto per ospitare una nuova sessione di gioco. Dovrebbe essere chiamata solo dopo aver completato la procedura di interruzione della sessione di gioco, perché segnala a Amazon GameLift Servers che il processo del server sia immediatamente disponibile per ospitare una nuova sessione di gioco.

Questa azione non viene richiamata se il processo del server verrà chiuso dopo l'interruzione della sessione di gioco. Invece, chiama ProcessEnding() per segnalare che sia la sessione di gioco che il processo sul server stanno terminando.

Sintassi

```
FGameLiftGenericOutcome TerminateGameSession()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

UpdatePlayerSessionCreationPolicy()

Aggiorna la capacità della sessione di gioco corrente di accettare nuove sessioni giocatore. Una sessione di gioco può essere configurata per accettare o rifiutare tutte le nuove sessioni giocatore. (Vedi anche l'`UpdateGameSession()`azione nella Amazon GameLift Servers Riferimento all'API del servizio).

Sintassi

```
FGameLiftGenericOutcome UpdatePlayerSessionCreationPolicy(EPlayerSessionCreationPolicy
 policy)
```

Parametri

Policy

Valore che indica se la sessione di gioco accetta nuovi giocatori.

Tipo: enum. `EPlayerSessionCreationPolicy`. I valori validi includono:

- ACCEPT_ALL: accetta tutte le nuove sessioni giocatore.
- DENY_ALL: rifiuta tutte le nuove sessioni giocatore.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Amazon GameLift ServersAmazon GameLift ServersriferimentoRealtime

Questa sezione contiene la documentazione di riferimento per l'Amazon GameLift ServersAmazon GameLift ServersRealtimeSDK. Include l'API Realtime Client e una guida per la configurazione dello script Amazon GameLift ServersRealtime.

Argomenti

- Amazon GameLift ServersRealtimeriferimento all'API client (C#)
- Riferimento allo script per Amazon GameLift ServersRealtime

Amazon GameLift ServersRealtimeriferimento all'API client (C#)

Usa l'API Realtime Client per preparare i client di gioco multiplayer da utilizzare con Amazon GameLift Servers Amazon GameLift ServersRealtime. L'API Client contiene un set di chiamate API sincrone e chiamate asincrone che consentono a un client di gioco di connettersi a un server Realtime e scambiare i messaggi e i dati con altri client tramite il server.

Questa API è definita nelle seguenti librerie:

Client.cs

- Operazioni sincrone
- Chiamate asincrone
- Tipi di dati

Per configurare l'API Realtime del client

1. Scarica l'SDK Amazon GameLift ServersRealtime del client.
2. Creare le librerie SDK C#. Prendere nota della posizione del file `GameLiftRealtimeClientSdkNet45.sln`. Consultare il file `README.md` per l'SDK del server per C# per i requisiti minimi e altre opzioni di creazione. In un IDE, caricare il file della soluzione. Per generare le librerie SDK, ripristina i NuGet pacchetti e crea la soluzione.
3. Aggiungi le librerie del client Realtime al tuo progetto del client di gioco.

Amazon GameLift ServersRealtimeriferimento all'API client (C#): Azioni

Questo riferimento all'API Realtime client C# può aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon GameLift Servers Realtime Deployed on fleets. Amazon GameLift Servers

- Operazioni sincrone
- Chiamate asincrone
- Tipi di dati

Client()

Inizializza un nuovo client per comunicare con il server Realtime e identifica il tipo di connessione da utilizzare.

Sintassi

```
public Client(ClientConfiguration configuration)
```

Parametri

clientConfiguration

Dettagli di configurazione che specificano il tipo di connessione. client/server Puoi decidere di chiamare Client() senza questo parametro, ma questo approccio produce una connessione non protetta per impostazione predefinita.

Tipo: ClientConfiguration

Campo obbligatorio: no

Valore restituito

Restituisce un'istanza del client Realtime da utilizzare con la comunicazione con il server Realtime.

Connect()

Richiede una connessione a un processo server che ospita una sessione di gioco.

Sintassi

```
public ConnectionStatus Connect(string endpoint, int remoteTcpPort, int listenPort,
 ConnectionToken token)
```

Parametri

endpoint

Nome DNS o indirizzo IP della sessione di gioco a cui connettersi. L'endpoint è specificato in un `GameSession` oggetto, che viene restituito in risposta a una chiamata del client alle azioni dell'Amazon GameLift ServersAPI AWS SDK StartGameSessionPlacement, CreateGameSessionoppure. DescribeGameSessions

> **Note**
>
> Se il server Realtime è in esecuzione su un parco istanze con un certificato TLS, è necessario utilizzare il nome DNS.

Tipo: stringa

Campo obbligatorio: sì

remoteTcpPort

Numero di porta per la connessione TCP assegnato alla sessione di gioco. Queste informazioni sono specificate in un `GameSession` oggetto, che viene restituito in risposta a una StartGameSessionPlacementCreateGameSessionrichiesta o DescribeGameSession.

Tipo: integer

Valori validi: da 1900 a 2000.

Campo obbligatorio: sì

listenPort

Numero di porta su cui il client di gioco è in ascolto per i messaggi inviati tramite il canale UDP.

Tipo: integer

Valori validi: da 33400 a 33500.

Campo obbligatorio: sì

token

Informazioni facoltative che identificano il client di gioco che sta eseguendo una richiesta al processo del server.

Tipo: ConnectionToken

Campo obbligatorio: sì

Valore restituito

Restituisce un valore ConnectionStatusenum che indica lo stato della connessione del client.

Disconnect()

Se connesso a una sessione di gioco, disconnette il client di gioco dalla sessione di gioco.

Sintassi

```
public void Disconnect()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Questo metodo non restituisce nulla.

NewMessage()

Crea un nuovo oggetto di messaggio con un codice operazione specifico. Quando l'oggetto messaggio viene restituito, completare il contenuto del messaggio specificando una destinazione, aggiornando il metodo di distribuzione e aggiungendo un payload di dati in base alle esigenze. Una volta completata questa operazione, inviare il messaggio utilizzando `SendMessage()`.

Sintassi

```
public RTMessage NewMessage(int opCode)
```

Parametri

opCode

Codice operativo definito dagli sviluppatori che identifica un evento di gioco o un'azione, ad esempio una mossa del giocatore o una notifica del server.

Tipo: integer

Campo obbligatorio: sì

Valore restituito

Restituisce un oggetto RTMessage contenente il codice dell'operazione specificata e il metodo di recapito predefinito. Per impostazione predefinita, il parametro delivery intent è impostato su `FAST`.

SendMessage()

Invia un messaggio a un giocatore o a un gruppo utilizzando il metodo di consegna specificato.

Sintassi

```
public void SendMessage(RTMessage message)
```

Parametri

message

Oggetto messaggio che specifica il destinatario target, il metodo di distribuzione e il contenuto del messaggio.

Tipo: RTMessage

Campo obbligatorio: sì

Valore restituito

Questo metodo non restituisce nulla.

JoinGroup()

Consente di aggiungere il giocatore come appartenente a un gruppo specificato. I gruppi possono contenere qualsiasi giocatore collegato al gioco. Una volta aggiunto, il giocatore riceve tutti i futuri messaggi inviati al gruppo e può inviare messaggi all'intero gruppo.

Sintassi

```
public void JoinGroup(int targetGroup)
```

Parametri

targetGroup

ID univoco che identifica il gruppo a cui aggiungere il giocatore. IDsI gruppi sono definiti dallo sviluppatore.

Tipo: integer

Campo obbligatorio: sì

Valore restituito

Questo metodo non restituisce nulla. Poiché questa richiesta viene inviata utilizzando il metodo di distribuzione affidabile (TCP), una richiesta non riuscita attiva una chiamata OnError().

LeaveGroup()

Consente di eliminare il giocatore come appartenente a un gruppo specifico. Se non è più parte del gruppo, il giocatore non riceve i messaggi inviati al gruppo e non può inviare messaggi all'intero gruppo.

Sintassi

```
public void LeaveGroup(int targetGroup)
```

Parametri

targetGroup

ID univoco che identifica il gruppo da cui aggiungere il giocatore. I gruppi IDs sono definiti dallo sviluppatore.

Tipo: integer

Campo obbligatorio: sì

Valore restituito

Questo metodo non restituisce nulla. Poiché questa richiesta viene inviata utilizzando il metodo di distribuzione affidabile (TCP), una richiesta non riuscita attiva una chiamata OnError().

RequestGroupMembership()

Richiede che un elenco di giocatori nel gruppo specificato venga inviato al client di gioco. Qualsiasi giocatore può richiedere queste informazioni, anche se non è un membro del gruppo. In risposta a questa richiesta, l'elenco di appartenenze viene inviato al client tramite una chiamata OnGroupMembershipUpdated().

Sintassi

```
public void RequestGroupMembership(int targetGroup)
```

Parametri

targetGroup

ID univoco che identifica il gruppo per ottenere le informazioni per la sottoscrizione. I gruppi IDs sono definiti dallo sviluppatore.

Tipo: integer

Campo obbligatorio: sì

Valore restituito

Questo metodo non restituisce nulla.

Amazon GameLift ServersRealtimeriferimento all'API client (C#): callback asincroni

Usa questo riferimento all'API Realtime del client C# per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon GameLift Servers Realtime Deployed on fleets. Amazon GameLift Servers

- Operazioni sincrone

- Chiamate asincrone
- Tipi di dati

Un client di gioco deve implementare questi metodi di chiamata per rispondere a eventi. Il server Realtime invoca queste chiamate per inviare le informazioni relative ai giochi al client di gioco. I metodi di chiamata per gli stessi eventi possono anche essere implementati con la logica di gioco personalizzato nello script del server Realtime. Consultare Richiamate di script per Amazon GameLift ServersRealtime.

I metodi di chiamata sono definiti nella `ClientEvents.cs`.

OnOpen()

Invocata quando il processo del server accetta la richiesta di connessione del client di gioco e apre una connessione.

Sintassi

```
public void OnOpen()
```

Parametri

Questo metodo non assume parametri.

Valore restituito

Questo metodo non restituisce nulla.

OnClose()

Invocata quando il processo del server termina la connessione con il client di gioco, ad esempio dopo la fine di una sessione di gioco.

Sintassi

```
public void OnClose()
```

Parametri

Questo metodo non assume parametri.

Valore restituito

Questo metodo non restituisce nulla.

OnError()

Richiamata quando si verifica un errore per una richiesta all'API del client Realtime. Questa chiamata può essere personalizzata per gestire un'ampia gamma di errori di connessione.

Sintassi

```
private void OnError(byte[] args)
```

Parametri

Questo metodo non assume parametri.

Valore restituito

Questo metodo non restituisce nulla.

OnDataReceived()

Richiamata quando il client di gioco riceve un messaggio dal server Realtime. Questo è il metodo principale con cui messaggi e notifiche vengono ricevuti da un client di gioco.

Sintassi

```
public void OnDataReceived(DataReceivedEventArgs dataReceivedEventArgs)
```

Parametri

dataReceivedEventArgomenti

Informazioni relative all'attività dei messaggi.

Tipo: DataReceivedEventArgs

Campo obbligatorio: sì

Valore restituito

Questo metodo non restituisce nulla.

OnGroupMembershipUpdated()

Richiamata quando la sottoscrizione per un gruppo a cui il giocatore appartiene è stata aggiornata. Questa chiamata viene anche invocata quando un client chiama `RequestGroupMembership`.

Sintassi

```
public void OnGroupMembershipUpdated(GroupMembershipEventArgs groupMembershipEventArgs)
```

Parametri

groupMembershipEventArghi

Le informazioni relative all'attività di appartenenza al gruppo.

Tipo: GroupMembershipEventArgs

Campo obbligatorio: sì

Valore restituito

Questo metodo non restituisce nulla.

Amazon GameLift ServersRealtimeriferimento all'API client (C#): tipi di dati

Questo riferimento all'API Realtime client C# può aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon GameLift Servers Realtime Deployed on fleets. Amazon GameLift Servers

- Operazioni sincrone
- Chiamate asincrone
- Tipi di dati

ClientConfiguration

Informazioni sul modo in cui il client di gioco si connette a un server Realtime.

Indice

ConnectionType

Tipo di client/server connessione da utilizzare, protetta o non protetta. Se non specifichi un tipo di connessione, l'impostazione predefinita corrisponde a una connessione non protetta.

> **Note**
>
> Quando ci si connette a un server Realtime su una flotta protetta con un certificato TLS, è necessario utilizzare il valore RT_OVER_WSS_DTLS_. TLS12

Tipo: un valore `ConnectionType` enum.

Campo obbligatorio: no

ConnectionToken

Informazioni sul giocatore del client di gioco che richiede una connessione con un server. and/or Realtime

Indice

playerSessionId

ID univoco rilasciato da Amazon GameLift Servers quando viene creata una nuova sessione di gioco. Un ID di sessione del giocatore viene specificato in un `PlayerSession` oggetto, che viene restituito in risposta a una chiamata del client alle azioni GameLift API StartGameSessionPlacement CreateGameSession, DescribeGameSessionPlacement, o DescribePlayerSessions.

Tipo: stringa

Campo obbligatorio: sì

payload

Informazioni definite dallo sviluppatore da comunicare al server Realtime al momento della connessione. Sono inclusi i dati arbitrari che potrebbero essere utilizzati per un meccanismo di accesso personalizzato. Ad esempio, un payload può fornire informazioni di autenticazione che lo script del server Realtime deve elaborare prima di consentire a un client di connettersi.

Tipo: array di byte

Campo obbligatorio: no

RTMessage

Informazioni su contenuti e distribuzione per un messaggio. Un messaggio deve specificare un giocatore o un gruppo di destinazione.

Indice

opCode

Codice operativo definito dagli sviluppatori che identifica un evento di gioco o un'azione, ad esempio una mossa del giocatore o una notifica del server. Il codice Op del messaggio fornisce contesto per il payload di dati fornito. Il codice operativo dei messaggi creati utilizzando NewMessage () è già impostato, ma può essere modificato in qualsiasi momento.

Tipo: integer

Campo obbligatorio: sì

targetPlayer

ID univoco che identifica il giocatore, cioè il destinatario previsto del messaggio inviato. Il target potrebbe essere il server stesso (utilizzando l'ID server) o un altro giocatore (utilizzando un ID giocatore).

Tipo: integer

Campo obbligatorio: no

targetGroup

ID univoco che identifica il gruppo, cioè il destinatario previsto del messaggio inviato. IDs I gruppi sono definiti dallo sviluppatore.

Tipo: integer

Campo obbligatorio: no

deliveryIntent

Indica se inviare il messaggio utilizzando la connessione affidabile TCP o utilizzando il canale veloce UDP. Messaggi creati utilizzando NewMessage().

Tipo: DeliveryIntent enum

Valori validi: FAST | RELIABLE

Campo obbligatorio: sì

payload

Contenuto del messaggio. Queste informazioni sono strutturate secondo quanto necessario per essere elaborate dal client di gioco in base al codice di operazione di accompagnamento. Può contenere i dati di gioco o altre informazioni che devono essere comunicate tra client di gioco o tra un client e il server di gioco Realtime.

Tipo: array di byte

Campo obbligatorio: no

DataReceivedEventArgs

Dati forniti con una chiamata OnDataReceived().

Indice

mittente

ID univoco che identifica l'entità (ID giocatore o ID server) che ha dato origine al messaggio.

Tipo: integer

Campo obbligatorio: sì

opCode

Codice operativo definito dagli sviluppatori che identifica un evento di gioco o un'azione, ad esempio una mossa del giocatore o una notifica del server. Il codice Op del messaggio fornisce contesto per il payload di dati fornito.

Tipo: integer

Campo obbligatorio: sì

dati

Contenuto del messaggio. Queste informazioni sono strutturate secondo quanto necessario per essere elaborate dal client di gioco in base al codice di operazione di accompagnamento. Può contenere i dati di gioco o altre informazioni che devono essere comunicate tra client di gioco o tra un client e il server di gioco Realtime.

Tipo: array di byte

Campo obbligatorio: no

GroupMembershipEventArgs

Dati forniti con una chiamata OnGroupMembershipUpdated().

Indice

mittente

ID univoco che identifica il giocatore che ha richiesto un aggiornamento delle sottoscrizioni al gruppo.

Tipo: integer

Campo obbligatorio: sì

opCode

Codice dell'operazione definito dagli sviluppatori che identifica un evento di gioco o un'azione.

Tipo: integer

Campo obbligatorio: sì

groupId

ID univoco che identifica il gruppo, cioè il destinatario previsto del messaggio inviato. IDs I gruppi sono definiti dallo sviluppatore.

Tipo: integer

Campo obbligatorio: sì

playerId

Elenco dei giocatori IDs che sono membri attuali del gruppo specificato.

Tipo: array di numeri interi

Campo obbligatorio: sì

enumerazioni;

Gli enum definiti per l'SDK del client Amazon GameLift Servers Realtime sono definiti come segue:

ConnectionStatus

- CONNESSO: il client di gioco è connesso al Realtime server solo tramite una connessione TCP. Tutti i messaggi indipendentemente dall'intento di distribuzione vengono inviati tramite TCP.
- CONNECTED_SEND_FAST — Il client di gioco è connesso al Realtime server tramite una connessione TCP e UDP. Tuttavia, la possibilità di ricevere messaggi tramite UDP non è ancora stata verificata e, di conseguenza, tutti i messaggi inviati al client di gioco usano TCP.
- CONNECTED_SEND_AND_RECEIVE_FAST — Il client di gioco è connesso al server tramite una connessione TCP e UDP. Realtime Il client di gioco è in grado di inviare e ricevere messaggi utilizzando UDP o TCP.
- CONNECTING Il client di gioco ha inviato una richiesta di connessione e il server Realtime la sta elaborando.
- DISCONNECTED_CLIENT_CALL — Il client di gioco è stato disconnesso dal server in risposta a una richiesta del client di gioco. Realtime Disconnect()
- DISCONNESSO: il client di gioco è stato disconnesso dal Realtime server per un motivo diverso da una chiamata di disconnessione del client.

ConnectionType

- TLS12 RT_OVER_WSS_DTLS_ — Tipo di connessione sicura.

  Da utilizzare con server in tempo reale in esecuzione su una flotta con un certificato TLS generato. Amazon GameLift Servers Quando si utilizza una connessione sicura, il traffico TCP viene crittografato con TLS 1.2 e il traffico UDP viene crittografato con DTLS 1.2.
- RT_OVER_WS_UDP_UNSECURED — Tipo di connessione non sicuro.
- RT_OVER_WEBSOCKET — Tipo di connessione non sicuro. Questo valore non è più preferito.

DeliveryIntent

- VELOCE: fornito utilizzando un canale UDP.
- AFFIDABILE: fornito utilizzando una connessione TCP.

Riferimento allo script per Amazon GameLift ServersRealtime

Utilizza queste risorse per creare una logica personalizzata nei tuoi script Realtime.

Argomenti

Richiamate di script per Amazon GameLift ServersRealtime

È possibile fornire una logica personalizzata per rispondere a eventi tramite l'implementazione di queste chiamate nello script Realtime.

Inizialo

Inizializza il server Realtime e riceve un'interfaccia del server Realtime.

Sintassi

```
init(rtsession)
```

onMessage

Invocato quando un messaggio ricevuto viene inviato al server.

Sintassi

```
onMessage(gameMessage)
```

onHealthCheck

Invocato per impostare lo stato dello stato delle sessioni di gioco. Per impostazione predefinita, lo stato di integrità è integro (o `true`). Questa chiamata può essere implementata per eseguire i controlli dello stato personalizzati e restituisce uno stato.

Sintassi

```
onHealthCheck()
```

onStartGameSessione

Invocato quando una nuova sessione di gioco viene avviata, con un oggetto della sessione di gioco superato.

Sintassi

```
onStartGameSession(session)
```

onProcessTerminate

Invocato quando il processo del server viene terminato dal servizio Amazon GameLift Servers. Ciò può fungere da trigger per terminare correttamente la sessione di gioco. Non è necessario effettuare la chiamata `processEnding()`.

Sintassi

```
onProcessTerminate()
```

onPlayerConnect

Invocato quando un giocatore richiede una connessione e ha superato la convalida iniziale.

Sintassi

```
onPlayerConnect(connectMessage)
```

onPlayerAccepted

Invocato quando viene accettata la connessione di un giocatore.

Sintassi

```
onPlayerAccepted(player)
```

onPlayerDisconnect

Invocato quando un giocatore si disconnette dalla sessione di gioco inviando una richiesta di disconnessione o con altri mezzi.

Sintassi

```
onPlayerDisconnect(peerId)
```

onProcessStarted

Invocato all'avvio di un processo server. Questa chiamata consente allo script di eseguire operazioni personalizzate di preparazione per l'hosting di una sessione di gioco.

Sintassi

```
onProcessStarted(args)
```

onSendToGiocatore

Invocato quando viene ricevuto un messaggio sul server da un giocatore che deve essere distribuito a un altro giocatore. Questo processo viene eseguito prima che il messaggio venga recapitato.

Sintassi

```
onSendToPlayer(gameMessage)
```

onSendToGruppo

Invocato quando viene ricevuto un messaggio sul server da un giocatore che deve essere distribuito a un gruppo. Questo processo viene eseguito prima che il messaggio venga recapitato.

Sintassi

```
onSendToGroup(gameMessage))
```

onPlayerJoinGruppo

Invocato quando un giocatore invia una richiesta per unirsi a un gruppo.

Sintassi

```
onPlayerJoinGroup(groupId, peerId)
```

onPlayerLeaveGruppo

Invocato quando un giocatore invia una richiesta per lasciare un gruppo.

Sintassi

```
onPlayerLeaveGroup(groupId, peerId)
```

Amazon GameLift ServersRealtimeinterfaccia

Quando uno Amazon GameLift Servers Realtime script viene inizializzato, viene restituita un'interfaccia al Realtime server. In questo argomento vengono illustrati i metodi e le proprietà disponibili tramite l'interfaccia. Per ulteriori informazioni sulla scrittura di script Realtime e per visualizzare un esempio di script dettagliato, consulta Creazione di uno Realtime script.

L'interfaccia di Realtime fornisce l'accesso ai seguenti oggetti:

- session
- player (giocatore)
- gameMessage
- configurazione

Oggetto session di Realtime

Utilizza questi metodi per accedere alle informazioni relative al server ed eseguire operazioni correlate al server.

getPlayers()

Recupera un elenco di peer IDs per i giocatori che sono attualmente connessi alla sessione di gioco. Restituisce un array di oggetti player.

Sintassi

```
rtSession.getPlayers()
```

broadcastGroupMembershipAggiorna ()

Attiva la distribuzione di un elenco aggiornato delle appartenenze ai gruppi di giocatori. Specificate quale iscrizione trasmettere (groupIdToBroadcast) e il gruppo a cui inviare l'aggiornamento (targetGroupId). Il gruppo IDs deve essere un numero intero positivo o «-1" per indicare tutti i gruppi. Vedi un Amazon GameLift ServersRealtimeesempio di script esempio di gruppo definito dall'utente. IDs

Sintassi

```
rtSession.broadcastGroupMembershipUpdate(groupIdToBroadcast, targetGroupId)
```

getServerId()

Recupera l'identificatore ID peer univoco del server, che viene utilizzato per instradare messaggi al server.

Sintassi

```
rtSession.getServerId()
```

getAllPlayersGroupId()

Recupera l'ID del gruppo per il gruppo predefinito che contiene tutti i giocatori attualmente connessi alla sessione di gioco.

Sintassi

```
rtSession.getAllPlayersGroupId()
```

processEnding()

Attiva il server Realtime per terminare il server di gioco. Questa funzione deve essere chiamata dallo script Realtime per terminare correttamente la sessione di gioco.

Sintassi

```
rtSession.processEnding()
```

getGameSessionId ()

Recupera l'ID univoco della sessione di gioco attualmente in esecuzione.

Sintassi

```
rtSession.getGameSessionId()
```

getLogger()

Recupera l'interfaccia per la creazione di log. Utilizza questo oggetto per le dichiarazioni di log che verranno acquisite nel log delle sessioni di gioco. Il logger supporta l'utilizzo delle dichiarazioni "info", "warn" e "error". Ad esempio: `logger.info("<string>")`.

Sintassi

```
rtSession.getLogger()
```

sendMessage()

Invia un messaggio, creato mediante `newTextGameMessage` oppure `newBinaryGameMessage` dal server Realtime a un destinatario giocatore utilizzando il canale UDP. Identifica il destinatario utilizzando l'ID peer del giocatore.

Sintassi

```
rtSession.sendMessage(gameMessage, targetPlayer)
```

sendGroupMessage()

Invia un messaggio, creato mediante `newTextGameMessage` oppure `newBinaryGameMessage` dal server Realtime a tutti i giocatori in un gruppo di giocatori utilizzando il canale UDP. Il gruppo IDs deve essere un numero intero positivo o «-1" per indicare tutti i gruppi. Vedi un Amazon GameLift ServersRealtimeesempio di script esempio di gruppo definito dall'utente. IDs

Sintassi

```
rtSession.sendGroupMessage(gameMessage, targetGroup)
```

sendReliableMessage()

Invia un messaggio, creato mediante `newTextGameMessage` oppure `newBinaryGameMessage` dal server Realtime a un destinatario giocatore utilizzando il canale TCP. Identifica il destinatario utilizzando l'ID peer del giocatore.

Sintassi

```
rtSession.sendReliableMessage(gameMessage, targetPlayer)
```

sendReliableGroupMessaggio ()

Invia un messaggio, creato mediante `newTextGameMessage` oppure `newBinaryGameMessage` dal server Realtime a tutti i giocatori in un gruppo di giocatori utilizzando il canale TCP. Gruppo IDs

che deve essere un numero intero positivo o «-1" per indicare tutti i gruppi. Vedi un Amazon GameLift ServersRealtimeesempio di script esempio di gruppo definito dall'utente. IDs

Sintassi

```
rtSession.sendReliableGroupMessage(gameMessage, targetGroup)
```

newTextGameMessaggio ()

Crea un nuovo messaggio contenente testo, da inviare dal server ai destinatari del giocatore utilizzando le SendMessage funzioni. Il formato del messaggio è simile al formato utilizzato nell'SDK del client per Realtime (vediRTMessage). Restituisce un oggetto `gameMessage`.

Sintassi

```
rtSession.newTextGameMessage(opcode, sender, payload)
```

newBinaryGameMessaggio ()

Crea un nuovo messaggio contenente dati binari, da inviare dal server ai destinatari del giocatore utilizzando le SendMessage funzioni. Il formato del messaggio è simile al formato utilizzato nell'SDK del client per Realtime (vediRTMessage). Restituisce un oggetto `gameMessage`.

Sintassi

```
rtSession.newBinaryGameMessage(opcode, sender, binaryPayload)
```

Oggetto player

Accedi alle informazioni sul giocatore.

player.peerId

L'ID univoco che viene assegnato a un client di gioco quando si connette al server Realtime e unito alla sessione di gioco.

giocatore. playerSessionId

L'ID della sessione del giocato Player ID della sessione viene fatto riferimento dal client di gioco quando è presente una connessione al server Realtime unito alla sessione di gioco.

Oggetto game message

Utilizza questi metodi per accedere ai messaggi che vengono ricevuti dal server Realtime. I messaggi ricevuti dai client di gioco hanno la struttura RTMessage.

getPayloadAsTesto ()

Ottiene il payload dei messaggi di gioco come testo.

Sintassi

```
gameMessage.getPayloadAsText()
```

gameMessage.opcode

Il codice dell'operazione contenuto in un messaggio.

gameMessage.payload

Il payload contenuto in un messaggio. Può essere testo o binario.

gameMessage.sender

L'ID peer del client di gioco che ha inviato un messaggio.

gameMessage.reliable

Il valore booleano che indica se il messaggio è stato inviato tramite TCP (true) o UDP (false).

Oggetto di configurazione

L'oggetto di configurazione può essere usato per sovrascrivere le configurazioni predefinite.

Configuration.maxPlayers

Il numero massimo di connessioni client/server che possono essere accettate da. RealTimeServers

Il valore predefinito è 32.

configurazione. pingIntervalTime

Intervallo di tempo in millisecondi in cui il server tenterà di inviare un ping a tutti i client connessi per verificare che le connessioni siano integre.

L'impostazione predefinita è 3000 ms.

Eventi di collocamento delle sessioni di gioco

Amazon GameLift Servers emette eventi per ogni richiesta di posizionamento della sessione di gioco man mano che viene elaborata. Puoi pubblicare questi eventi su un argomento di Amazon SNS, come descritto in. Imposta la notifica degli eventi per il posizionamento della sessione di gioco Questi eventi vengono inoltre trasmessi ad Amazon CloudWatch Events quasi in tempo reale e con la massima diligenza possibile.

Questo argomento descrive la struttura degli eventi di collocamento delle sessioni di gioco e fornisce un esempio per ogni tipo di evento. Per ulteriori informazioni sullo stato delle richieste di collocamento delle sessioni di gioco, GameSessionPlacementconsulta la Amazon GameLift Servers Riferimento API.

Sintassi degli eventi di posizionamento

Gli eventi sono rappresentati come oggetti JSON. La struttura degli eventi è conforme al modello CloudWatch Events, con campi di primo livello simili e dettagli specifici del servizio.

I campi di primo livello includono quanto segue (vedi Event Pattern per maggiori dettagli):

version

Questo campo è sempre impostato su 0 (zero).

id

Identificatore di tracciamento univoco per l'evento.

detail-type (tipo di dettaglio)

Il valore è sempre`GameLift Queue Placement Event`.

source

Il valore è sempre`aws.gamelift`.

account

L' AWS account che viene utilizzato per gestire Amazon GameLift Servers.

time

Timestamp dell'evento.

Regione

La AWS regione in cui viene elaborata la richiesta di collocamento. Questa è la regione in cui si trova la coda della sessione di gioco in uso.

risorse

Valore ARN della coda della sessione di gioco che sta elaborando la richiesta di posizionamento.

PlacementFulfilled

La richiesta di collocamento è stata soddisfatta con successo. È stata avviata una nuova sessione di gioco e sono state create nuove sessioni di gioco per ogni giocatore elencato nella richiesta di posizionamento della sessione di gioco. Le informazioni sulla connessione del giocatore sono disponibili.

Sintassi dettagliata:

PlacementID

Un identificatore univoco assegnato alla richiesta di posizionamento della sessione di gioco.

port

Il numero di porta per la nuova sessione di gioco.

gameSessionArn

L'identificatore ARN per la nuova sessione di gioco.

ipAddress

L'indirizzo IP della sessione di gioco.

Nome DNS

L'identificatore DNS assegnato all'istanza che esegue la nuova sessione di gioco. Il formato dei valori è diverso a seconda che l'istanza che esegue la sessione di gioco sia abilitata per TLS. Quando si connettono a una sessione di gioco su una flotta che supporta TLS, i giocatori devono utilizzare il nome DNS, non l'indirizzo IP.

Flotte abilitate per TLS:. `<unique identifier>.<region identifier>.amazongamelift.com`

Non-TLS-enabled flotte:. `ec2-<unique identifier>.compute.amazonaws.com`

startTime

Timestamp che indica quando questa richiesta è stata inserita nella coda.

endTime

Timestamp che indica quando questa richiesta è stata soddisfatta.

gameSessionRegion

AWS Regione della flotta che ospita la sessione di gioco. Corrisponde al token della regione in`GameSessionArn`.

placedPlayerSessions

La raccolta di sessioni dei giocatori che sono state create per ogni giocatore nella richiesta di posizionamento della sessione di gioco.

Esempio

```
{
  "version": "0",
  "id": "1111aaaa-bb22-cc33-dd44-5555eeee66ff",
  "detail-type": "GameLift Queue Placement Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2021-03-01T15:50:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:gamesessionqueue/MegaFrogRace-NA"
  ],
  "detail": {
    "type": "PlacementFulfilled",
    "placementId": "9999ffff-88ee-77dd-66cc-5555bb44aa",
    "port": "6262",
    "gameSessionArn": "arn:aws:gamelift:us-west-2::gamesession/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa/4444dddd-55ee-66ff-77aa-8888bbbb99cc",
    "ipAddress": "98.987.98.987",
    "dnsName": "ec2-12-345-67-890.us-west-2.compute.amazonaws.com",
    "startTime": "2021-03-01T15:50:49.741Z",
    "endTime": "2021-03-01T15:50:52.084Z",
    "gameSessionRegion": "us-west-2",
    "placedPlayerSessions": [
      {
```

```
        "playerId": "player-1"
        "playerSessionId": "psess-123213123232412412312 3"
      }
    ]
  }
}
```

PlacementCancelled

La richiesta di collocamento è stata annullata con una chiamata al GameLift servizio. StopGameSessionPlacement

Dettaglio:

ID di posizionamento

Un identificatore univoco assegnato alla richiesta di posizionamento della sessione di gioco.

startTime

Timestamp che indica quando questa richiesta è stata inserita nella coda.

endTime

Timestamp che indica quando questa richiesta è stata annullata.

Esempio

```
{
  "version": "0",
  "id": "1111aaaa-bb22-cc33-dd44-5555eeee66ff",
  "detail-type": "GameLift Queue Placement Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2021-03-01T15:50:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:gamesessionqueue/MegaFrogRace-NA"
  ],
  "detail": {

    "type": "PlacementCancelled",
    "placementId": "9999ffff-88ee-77dd-66cc-5555bb44aa",
    "startTime": "2021-03-01T15:50:49.741Z",
```

```
    "endTime": "2021-03-01T15:50:52.084Z"
  }
}
```

PlacementTimedOut

Il posizionamento della sessione di gioco non è stato completato con successo prima della scadenza del limite di tempo della coda. La richiesta di piazzamento può essere inoltrata nuovamente se necessario.

Dettaglio:

ID di posizionamento

Un identificatore univoco assegnato alla richiesta di posizionamento della sessione di gioco.

startTime

Timestamp che indica quando questa richiesta è stata inserita nella coda.

endTime

Timestamp che indica quando questa richiesta è stata annullata.

Esempio

```
{
  "version": "0",
  "id": "1111aaaa-bb22-cc33-dd44-5555eeee66ff",
  "detail-type": "GameLift Queue Placement Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2021-03-01T15:50:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:gamesessionqueue/MegaFrogRace-NA"
  ],
  "detail": {

    "type": "PlacementTimedOut",
    "placementId": "9999ffff-88ee-77dd-66cc-5555bb44aa",
    "startTime": "2021-03-01T15:50:49.741Z",
    "endTime": "2021-03-01T15:50:52.084Z"
```

```
  }
}
```

PlacementFailed

Amazon GameLift Servers non è stato in grado di soddisfare la richiesta di sessione di gioco. Ciò è generalmente causato da un errore interno imprevisto. La richiesta di collocamento può essere inoltrata nuovamente se necessario.

Dettaglio:

ID di posizionamento

Un identificatore univoco assegnato alla richiesta di posizionamento della sessione di gioco.

startTime

Timestamp che indica quando questa richiesta è stata inserita nella coda.

endTime

Timestamp che indica quando questa richiesta non è riuscita.

Esempio

```
{
  "version": "0",
  "id": "39c978f3-ba46-3f7c-e787-55bfcca1bd31",
  "detail-type": "GameLift Queue Placement Event",
  "source": "aws.gamelift",
  "account": "252386620677",
  "time": "2021-03-01T15:50:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:gamelift:us-west-2:252386620677:gamesessionqueue/MegaFrogRace-NA"
  ],
  "detail": {

    "type": "PlacementFailed",
    "placementId": "e4a1119a-39af-45cf-a990-ef150fe0d453",
    "startTime": "2021-03-01T15:50:49.741Z",
    "endTime": "2021-03-01T15:50:52.084Z"
  }
```

```
}
```

Amazon GameLift ServersVersioni AMI

La tabella seguente identifica le immagini di macchine Amazon più recenti (AMIs) Amazon GameLift Servers utilizzate per l' EC2 hosting gestito. Come descritto inAnalisi della configurazione e delle vulnerabilità in Amazon GameLift Servers, è necessario creare regolarmente nuove EC2 flotte Amazon GameLift Servers gestite per distribuire gli ultimi aggiornamenti delle versioni AMI.

AMIs da utilizzare con server SDK for 5+ Amazon GameLift Servers

Amazon GameLift Serversutilizza quanto segue AMIs per ospitare server di gioco integrati con il server SDK per Amazon GameLift Servers la versione 5.

Amazon GameLift Serversimmagine	Architettura	Patch più recente	AWS Immagine base
Amazon Linux 2023 BASE_AMI_LINUX_2023	x86	2025-06-12	Amazon Linux 2023 AMI 2023.7.20 250609.0 x86_64 HVM kernel-6.1 (note di rilascio)
Amazon Linux 2 BASE_AMI_LINUX_2_SDK_5	x86	2025-06-12	AMI Amazon Linux 2 Kernel 5.10 2.0.20250603.0 x86_64 HVM gp2 (note di rilascio)
Windows 2016 BASE_AMI_WINDOWS_2016_SDK_5	x86	2025-06-12	Windows_Server-2016-Inglese-Full-Base-2025.06.12 (note di rilascio)
Amazon Linux 2023 BASE_AMI_LINUX_2023_ARM	ARM64	2025-06-12	Amazon Linux 2023 AMI 2023.7.20 250609.0 arm64 HVM kernel-6.1 (note di rilascio)
Amazon Linux 2	ARM64	2025-06-12	AMI Amazon Linux 2 LTS Arm64 Kernel 5.10 2.0.20250603.0 arm64 HVM gp2 (note di rilascio)

Amazon GameLift Serversimmagine	Architettura	Patch più recente	AWS Immagine base
BASE_AMI_LINUX_2_ARM			

AMIs da utilizzare Amazon GameLift Servers con server SDK for 4

Amazon GameLift Serversutilizza quanto segue AMIs per ospitare server di gioco integrati con server SDK per la Amazon GameLift Servers versione 4 o precedente.

Amazon GameLift Serversimmagine	Architettura	Patch più recente	AWS Immagine base
Amazon Linux 2 BASE_AMI_LINUX_2_SDK_4	x86	2025-06-12	AMI Amazon Linux 2 2.0.20250605.0 x86_64 HVM gp2 (note di rilascio)
Windows 2016 BASE_AMI_WINDOWS_2016_SDK_4	x86	2025-06-12	Windows_Server-2016-Inglese-Full-Base-2025.06.12 (note di rilascio)

Per ulteriori informazioni, consulta le seguenti risorse:

- Note di rilascio di Amazon Linux 2023
- Note di rilascio di Amazon Linux 2
- AWS Cronologia delle versioni di Windows AMI
- Descrizione delle modifiche ai contenuti di Software Update Services e Windows Server Update Services per il 2024

Endpoint e quote di Amazon GameLift Servers

- Per gli endpoint di servizio e gli endpoint beacon che puoi utilizzare per connetterti programmaticamente al servizio, consulta Service Endpoints. Amazon GameLift Servers

- Per le Amazon GameLift Servers quote sull'uso delle risorse o delle operazioni del servizio per account, vedi Service Quotas. AWSAmazon GameLift Servers Per ulteriori informazioni sulle quote e su come richiedere un aumento, consulta AWS le quote di servizio. Puoi richiedere aumenti delle quote utilizzando la Amazon GameLift Servers console.

Amazon GameLift ServersFari ping UDP

I beacon ping UDP forniscono un modo per misurare la latenza di rete tra i dispositivi dei giocatori e le posizioni di hosting. Amazon GameLift Servers Con i ping beacon, puoi raccogliere dati di latenza accurati per prendere decisioni informate sul posizionamento dei server di gioco e migliorare il matchmaking dei giocatori in base ai requisiti di latenza.

Come funzionano i ping beacon UDP

Amazon GameLift Serversfornisce endpoint UDP fissi (ping beacon) in ogni posizione di hosting in cui è possibile distribuire server di gioco. Poiché la maggior parte dei server di gioco comunica tramite UDP, la misurazione della latenza con i beacon ping UDP fornisce risultati più accurati rispetto all'utilizzo dei ping ICMP. I dispositivi di rete spesso gestiscono i pacchetti ICMP in modo diverso dai pacchetti UDP, il che può portare a misurazioni della latenza che non riflettono le prestazioni reali dei giocatori.

Con i ping beacon UDP, il tuo client di gioco può inviare messaggi UDP a questi endpoint e ricevere risposte asincrone, ottenendo misurazioni della latenza che rappresentano meglio le condizioni effettive del traffico di gioco tra il dispositivo di un giocatore e le potenziali località di hosting. Gli endpoint sono permanenti e rimangono disponibili fintanto che supporta l'hosting di giochi in quella posizione. Amazon GameLift Servers

Casi d'uso comuni dei beacon ping UDP

Puoi utilizzare i beacon ping UDP in diversi modi per ottimizzare l'esperienza di rete del gioco.

Scelta delle location di hosting ottimali

Raccogli i dati sulla latenza in diverse aree geografiche per identificare le migliori posizioni primarie e di backup per ospitare i server di gioco per la tua base di giocatori.

Posizionamento delle sessioni di gioco in base alla latenza dei giocatori

Includi i dati sulla latenza dei giocatori quando richiedi nuove sessioni di gioco per aiutarti a scegliere le località che offrono l'esperienza di latenza più bassa.

Ottimizzazione del matchmaking in base alla latenza

Fornisci dati sulla latenza dei giocatori quando richiedi il matchmaking per aiutare ad abbinare giocatori con profili di latenza simili e a collocare le sessioni di gioco in posizioni ottimali per i giocatori abbinati.

> **Note**
>
> Quando crei richieste di matchmaking, non dovresti fornire informazioni sulla latenza a località in cui non disponi di flotte. In tal caso, Amazon GameLift Servers potresti provare a collocare la sessione di gioco in luoghi in cui la flotta non è disponibile, con conseguenti errori nelle richieste di matchmaking.

Ottenere endpoint beacon

Per recuperare le informazioni sul dominio e sulla porta del ping beacon per le Amazon GameLift Servers posizioni, utilizza l'operazione API. ListLocations Il set di posizioni restituito da questa API dipende dalla regione specificata al Regione AWS momento della chiamata (o dalla regione predefinita se non ne specifichi una). Quando chiami da:

- Una regione di origine di una flotta che supporta più sedi: l'API restituisce informazioni per tutte le sedi di hosting
- Una regione di origine di un parco veicoli che supporta un'unica ubicazione: l'API restituisce le informazioni relative a quella località

Tieni presente che se chiami questa API utilizzando una posizione che può essere solo una posizione remota in un parco di sedi multiple, l'API restituirà un errore perché quel tipo di posizione non dispone di un endpoint di servizio.

Consulta la tabella delle località supportate in AWS Località supportate per identificare le aree geografiche che supportano flotte con una o più sedi.

Esempio

```
aws gamelift list-locations --region ap-northeast-2
```

Questo Regione AWS supporta flotte con più sedi, pertanto verranno restituite più sedi. Ecco un esempio di uno dei valori restituiti:

```
[...]
{
    "LocationName": "ap-northeast-1",
    "PingBeacon": {
        "UDPEndpoint": {
            "Domain": "gamelift-ping.ap-northeast-1.api.aws",
            "Port": 7770
        }
    }
}
```

> **Important**
>
> Memorizza nella cache le informazioni del beacon ping anziché chiamare `ListLocations` prima di ogni misurazione della latenza. Le informazioni sul dominio e sulla porta sono statiche e l'API non è progettata per richieste ad alto volume.

Implementazione della misurazione della latenza

Segui queste best practice quando implementi le misurazioni della latenza utilizzando i beacon ping UDP:

1. Memorizza le informazioni sui ping beacon utilizzando uno di questi approcci:
   - Codifica gli endpoint nel tuo client di gioco.
   - Memorizza le informazioni nel backend del gioco.
   - Implementa un meccanismo di aggiornamento periodico (giornaliero/settimanale) per aggiornare le informazioni.
2. Invia messaggi ping UDP:
   - Inserisci quello che vuoi nel corpo del messaggio, purché non sia vuoto, e mantieni i messaggi al di sotto della dimensione massima di 300 byte.
   - Rispetta i seguenti limiti di velocità per ogni località:
     - 3 transazioni al secondo (TPS) per combinazione univoca di indirizzo IP e porta del mittente
     - 1000 TPS per indirizzo IP univoco del mittente
3. Calcola la latenza:
   - Invia più ping a ciascuna posizione per calcolare una latenza media.

- Valuta la possibilità di inviare ping simultanei a più sedi per risultati più rapidi.
- Se necessario, utilizzate la logica di ripetizione per inviare nuovi pacchetti per qualsiasi pacchetto che non sia stato restituito entro breve tempo (in genere 1-3 secondi), poiché UDP non offre una consegna garantita al 100%.
- Calcola la differenza di tempo tra l'invio di un messaggio e la ricezione della risposta.
- Se una gran parte dei ping UDP verso una posizione non ottiene costantemente una risposta, calcola la latenza utilizzando i ping ICMP ai nostri endpoint di Amazon GameLift Serversservizio standard come fallback.

> **Tip**
>
> Ti consigliamo di includere la porta 7770 ovunque documenti l'elenco delle porte che i giocatori devono avere aperte sulla rete locale. Questo è un altro motivo per cui dovresti disporre di un sistema di riserva per misurare la latenza (usando ICMP, ad esempio) nel caso in cui questa porta sia bloccata.

Esempi di codice

Ecco alcuni semplici esempi che mostrano come inviare ping UDP e calcolare la latenza.

C++

```
#include <iostream>
#include <cstring>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <unistd.h>
#include <chrono>

int main() {
    // Replace with Amazon GameLift Servers UDP ping beacon domain for your desired
 location
    const char* domain = "gamelift-ping.ap-south-1.api.aws";
    const int port = 7770;
    const char* message = "Ping";        // Your message
    const int num_pings = 3;            // Number of pings to send
```

```
    // Create socket
    int sock = socket(AF_INET, SOCK_DGRAM, 0);
    if (sock < 0) {
        std::cerr << "Error creating socket" << std::endl;
        return 1;
    }

    // Resolve domain name to IP address
    struct hostent* host = gethostbyname(domain);
    if (host == nullptr) {
        std::cerr << "Error resolving hostname" << std::endl;
        close(sock);
        return 1;
    }

    // Set up the server address structure
    struct sockaddr_in server_addr;
    std::memset(&server_addr, 0, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(port);
    std::memcpy(&server_addr.sin_addr, host->h_addr, host->h_length);

    // Set socket timeout
    struct timeval tv;
    tv.tv_sec = 1;  // 1 second timeout
    tv.tv_usec = 0;
    if (setsockopt(sock, SOL_SOCKET, SO_RCVTIMEO, &tv, sizeof(tv)) < 0) {
        std::cerr << "Error setting socket timeout" << std::endl;
        close(sock);
        return 1;
    }

    double total_latency = 0;
    int successful_pings = 0;

    for (int i = 0; i < num_pings; ++i) {
        auto start = std::chrono::high_resolution_clock::now();

        // Send the message
        ssize_t bytes_sent = sendto(sock, message, std::strlen(message), 0,
                                    (struct sockaddr*)&server_addr,
sizeof(server_addr));
```

```
        if (bytes_sent < 0) {
            std::cerr << "Error sending message" << std::endl;
            continue;
        }

        // Receive response
        char buffer[1024];
        socklen_t server_addr_len = sizeof(server_addr);
        ssize_t bytes_received = recvfrom(sock, buffer, sizeof(buffer), 0,
                                          (struct sockaddr*)&server_addr,
 &server_addr_len);

        auto end = std::chrono::high_resolution_clock::now();

        if (bytes_received < 0) {
            std::cerr << "Error receiving response or timeout" << std::endl;
        } else {
            std::chrono::duration<double, std::milli> latency = end - start;
            total_latency += latency.count();
            successful_pings++;

            std::cout << "Received response, latency: " << latency.count() << " ms"
 << std::endl;
        }

        // Wait a bit before next ping
        usleep(1000000);  // 1s
    }

    // Close the socket
    close(sock);

    if (successful_pings > 0) {
        double avg_latency = total_latency / successful_pings;
        std::cout << "Average latency: " << avg_latency << " ms" << std::endl;
    } else {
        std::cout << "No successful pings" << std::endl;
    }

    return 0;
}
```

C#

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Diagnostics;
using System.Threading.Tasks;

class UdpLatencyTest
{
    static async Task Main()
    {
        // Replace with Amazon GameLift Servers UDP ping beacon domain for your
 desired location
        string domain = "gamelift-ping.ap-south-1.api.aws";
        int port = 7770;
        string message = "Ping";         // Your message
        int numPings = 3;                // Number of pings to send
        int timeoutMs = 1000;            // Timeout in milliseconds

        await MeasureLatency(domain, port, message, numPings, timeoutMs);
    }

    static async Task MeasureLatency(string domain, int port, string message, int
 numPings, int timeoutMs)
    {
        using (var udpClient = new UdpClient())
        {
            try
            {
                // Resolve domain name to IP address
                IPAddress[] addresses = await Dns.GetHostAddressesAsync(domain);
                if (addresses.Length == 0)
                {
                    Console.WriteLine("Could not resolve domain name.");
                    return;
                }

                IPEndPoint endPoint = new IPEndPoint(addresses[0], port);
                byte[] messageBytes = Encoding.UTF8.GetBytes(message);

                // Set receive timeout
                udpClient.Client.ReceiveTimeout = timeoutMs;
```

```
                double totalLatency = 0;
                int successfulPings = 0;
                var stopwatch = new Stopwatch();

                for (int i = 0; i < numPings; i++)
                {
                    try
                    {
                        stopwatch.Restart();

                        // Send message
                        await udpClient.SendAsync(messageBytes, messageBytes.Length,
endPoint);

                        // Wait for response
                        UdpReceiveResult result = await
ReceiveWithTimeoutAsync(udpClient, timeoutMs);

                        stopwatch.Stop();
                        double latency = stopwatch.Elapsed.TotalMilliseconds;

                        totalLatency += latency;
                        successfulPings++;

                        string response = Encoding.UTF8.GetString(result.Buffer);
                        Console.WriteLine($"Ping {i + 1}: {latency:F2}ms - Response:
{response}");
                    }
                    catch (SocketException ex)
                    {
                        Console.WriteLine($"Ping {i + 1}: Failed - {ex.Message}");
                    }
                    catch (TimeoutException)
                    {
                        Console.WriteLine($"Ping {i + 1}: Timeout");
                    }

                    // Wait before next ping
                    await Task.Delay(1000); // 1s between pings
                }

                if (successfulPings > 0)
                {
```

```
                    double averageLatency = totalLatency / successfulPings;
                    Console.WriteLine($"\nSummary:");
                    Console.WriteLine($"Successful pings: {successfulPings}/
{numPings}");
                    Console.WriteLine($"Average latency: {averageLatency:F2}ms");
                }
                else
                {
                    Console.WriteLine("\nNo successful pings");
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Error: {ex.Message}");
            }
        }
    }

    static async Task<UdpReceiveResult> ReceiveWithTimeoutAsync(UdpClient client,
 int timeoutMs)
    {
        using var cts = new System.Threading.CancellationTokenSource(timeoutMs);
        try
        {
            return await client.ReceiveAsync().WaitAsync(cts.Token);
        }
        catch (OperationCanceledException)
        {
            throw new TimeoutException("Receive operation timed out");
        }
    }
}
```

Python

```
import socket
import time
import statistics
from datetime import datetime

def udp_ping(host, port, timeout=2):
    """
    Send a UDP ping and return the round trip time in milliseconds.
```

```
    Returns None if timeout occurs.
    """
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.settimeout(timeout)
    message = b'ping'

    try:
        # Resolve hostname first
        try:
            socket.gethostbyname(host)
        except socket.gaierror as e:
            print(f"Could not resolve hostname: {e}")
            return None

        start_time = time.time()
        sock.sendto(message, (host, port))

        # Wait for response
        data, server = sock.recvfrom(1024)
        end_time = time.time()

        # Calculate round trip time in milliseconds
        rtt = (end_time - start_time) * 1000
        return rtt

    except socket.timeout:
        print(f"Request timed out")
        return None
    except Exception as e:
        print(f"Error: {type(e).__name__}: {e}")
        return None
    finally:
        sock.close()

def main():
    # Replace with Amazon GameLift Servers UDP ping beacon domain for your desired
 location
    host = "gamelift-ping.ap-south-1.api.aws"
    port = 7770
    num_pings = 3
    latencies = []

    print(f"\nPinging {host}:{port} {num_pings} times...")
    print(f"Start time: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}\n")
```

```
    for i in range(num_pings):
        print(f"Ping {i+1}:")
        rtt = udp_ping(host, port)

        if rtt is not None:
            print(f"Response from {host}: time={rtt:.2f}ms")
            latencies.append(rtt)

        # Wait 1 second between pings
        if i < num_pings - 1:
            time.sleep(1)
        print()

    # Calculate and display statistics
    print("-" * 50)
    print(f"Ping statistics for {host}:")
    print(f"    Packets: Sent = {num_pings}, Received = {len(latencies)}, "
          f"Lost = {num_pings - len(latencies)} "
          f"({((num_pings - len(latencies)) / num_pings * 100):.1f}% loss)")

    if latencies:
        print("\nRound-trip latency statistics:")
        print(f"    Minimum = {min(latencies):.2f}ms")
        print(f"    Maximum = {max(latencies):.2f}ms")
        print(f"    Average = {statistics.mean(latencies):.2f}ms")

    print(f"\nEnd time: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}")

if __name__ == "__main__":
    main()
```

Note di rilascio di Amazon GameLift Servers

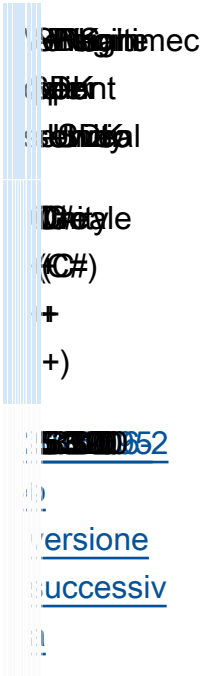
Le note di rilascio di Amazon GameLift Servers forniscono informazioni sulle nuove funzionalità, aggiornamenti e correzioni relative al servizio.

Versioni SDK

Le tabelle seguenti elencano tutte le Amazon GameLift Servers versioni con informazioni sulla versione dell'SDK. Non è necessario utilizzare strumenti simili SDKs per le integrazioni tra server di gioco e client. Tuttavia, le versioni precedenti di un SDK potrebbero non supportare completamente le funzionalità più recenti di un altro SDK.

Amazon GameLift Servers SDKs e i plugin sono open source. Consultare Ottieni strumenti Amazon GameLift Servers di sviluppo. Per ottenere le versioni più recenti, consulta l'Amazon GameLift Servers GitHub organizzazione.

Versione attuale



Versioni precedenti

- imec
- nt
- al
- o
- nity
- +
- Unreal
- 52
  - ersione
  - uccessiv
    -
- 52
  - .NET
  - rsione
  - uccessiv
    -
- 52
  - ersione
  - uccessiv
    -
- 51
  - ersione
  - uccessiv
    -
- 51
  - ersione

imec
nt
al

o
nity
+
Unreal

uccessiv

170

ersione
uccessiv

251

uccessiv

151

ersione
uccessiv

251

uccessiv

251

uccessiv

imec
nt
al

o
nity
+
Unreal

251
ersione
uccessiv
l

131
uccessiv

130
uccessiv

042
ersione
uccessiv
l

041
ersione
uccessiv
l

imec
nt
al

o
nity
+
Unreal

12
ersione
uccessiv

12
uccessiv

6-1
ersione
uccessiv

6-2
ersione
uccessiv

4-2
ersione
uccessiv

imec
nt
al

o
nity
+
Unreal

4-1
uccessiv

2-0
uccessiv

1-3
uccessiv

2-0
uccessiv

8-2
ersione
uccessiv

imec
nt
al

nity
+
Unreal

-2
ersione
uccessiv

-0
ersione
uccessiv

-2
ersione
uccessiv

-1
ersione
uccessiv

-0
uccessiv

imec
nt
al

o
nity
+
Unreal

2-2
ersione
uccessiv

01-2
ersione
uccessiv

01-1
uccessiv

09-1
uccessiv

08-2
ersione
uccessiv

imec
nt
al

o
nity
+
Unreal

-1

ersione
uccessiv
a

-0

ersione
uccessiv
a

-1

ersione
uccessiv
a

-1

ersione
uccessiv
a

001

ersione
uccessiv
a

imec
nt
al

o
nity
+
Unreal

001
ersione
uccessiv
a

001
ersione
uccessiv
a

04-2
ersione
uccessiv
a

03-0
ersione
uccessiv
a

02-0
ersione
uccessiv
a

imec
nt
al

o
nity
+
Unreal

12-1
ersione
uccessiv
a

09-2
ersione
uccessiv
a

06-1
ersione
uccessiv
a

05-1
ersione
uccessiv
a

02-1
ersione
uccessiv
a

imec
nt
al

o
nity
+
Unreal

02-0
ersione
uccessiv

09-0
ersione
uccessiv

08-1
ersione
uccessiv

1
ersione
uccessiv

4-1
ersione
uccessiv

imec
nt
al

o
nity
+
Unreal

02-2
ersione
uccessiv
a

11-1
ersione
uccessiv
a

10-1
ersione
uccessiv
a

09-0
ersione
uccessiv
a

08-0
ersione
uccessiv
a

Note di rilascio

Le seguenti note di rilascio sono in ordine cronologico, con gli aggiornamenti più recenti elencati per primi. Amazon GameLift Servers è stato rilasciato per la prima volta nel 2016. Per le note di rilascio precedenti a quelle elencate qui, consulta i collegamenti per le date di rilascio in Versioni SDK.

24 giugno 2025: Amazon GameLift Servers lancia i beacon ping UDP per aiutare a misurare la latenza di rete in tempo reale per i giochi multiplayer

Versioni SDK aggiornate:

- SDK AWS 1.11.595

Amazon GameLift Serversrilascia per la disponibilità generale una serie di endpoint fissi chiamati ping beacon UDP per aiutarti a misurare con precisione la latenza tra i dispositivi dei giocatori e le posizioni dei server di gioco. Gli endpoint dei beacon ping UDP sono disponibili in tutte le regioni AWS globali e le Local Zones supportate daAmazon GameLift Servers, ad eccezione delle regioni della Cina. AWS

La maggior parte dei giochi multiplayer utilizza UDP (User Datagram Protocol) come protocollo di trasmissione di pacchetti principale, grazie ai vantaggi in termini di prestazioni per i giochi in tempo reale. Comprendere e ottimizzare la latenza di rete è fondamentale per offrire la migliore esperienza di gioco possibile. I ping beacon UDP forniscono un modo coerente e affidabile per misurare la latenza effettiva dei pacchetti UDP tra giocatori e server di gioco, aiutandoti a prendere decisioni migliori sugli abbinamenti e sul posizionamento delle sessioni di gioco. player-to-server

L'ListLocationsAPI è stata estesa per includere le informazioni sul dominio e sulla porta degli endpoint come parte dell'elenco delle posizioni restituite, semplificando l'accesso programmatico agli endpoint.

Il tuo client di gioco può inviare messaggi UDP a questi endpoint e ricevere risposte asincrone con gli stessi dati, fornendoti misurazioni della latenza che rappresentano meglio le condizioni effettive del traffico di gioco tra il dispositivo di un giocatore e le potenziali località di hosting. Questi endpoint sono permanenti e rimangono disponibili fintanto che supporta l'hosting di giochi in quella località. Amazon GameLift Servers

Ulteriori informazioni:

- Fari di ping UDP, Guida per Amazon GameLift Serversgli sviluppatori

- ListLocations, Riferimento Amazon GameLift Servers API

24 giugno 2025: Amazon GameLift Servers aggiunge il supporto di localizzazione per Thailandia e Malesia

Con l'hosting Amazon GameLift Servers gestito, ora puoi distribuire risorse di server di gioco a Bangkok, in Thailandia, e Kuala Lumpur, in Malesia, estendendo la portata dei tuoi giochi ai giocatori di tutto il sud-est asiatico. Queste nuove regioni aiutano a ridurre la latenza e a migliorare l'esperienza di gioco dei giocatori in quelle aree.

Le seguenti Regioni AWS sono disponibili come postazioni remote per flotte con più sedi. Per iniziare a ospitare sessioni di gioco in queste località, aggiungile come postazioni remote a una flotta con più sedi nuova o esistente. Con le flotte con più sedi, puoi gestire direttamente la capacità di hosting in ogni sede.

- Asia Pacifico (Tailandia) `ap-southeast-7` ()
- Asia Pacifico (Malesia) () `ap-southeast-5`

Questi non Regioni AWS sono abilitati per impostazione predefinita per un AWS account. È necessario iscriversi a ciascuna regione prima di potervi distribuire Amazon GameLift Servers risorse.

Ulteriori informazioni:

- Amazon GameLift Serverssedi di assistenza, Guida per sviluppatori Amazon GameLift Serversdi hosting
- Aggiorna le posizioni della flotta, Guida per sviluppatori di Amazon GameLift Servers hosting
- Abilita o disabilita AWS le regioni nel tuo AWS account, Guida di riferimento per la gestione degli account.

29 maggio 2025: Amazon GameLift Servers rilascia nuove versioni dell'SDK del server e dei plugin per Unreal Engine e Unity

Versioni SDK aggiornate:

- SDK del server C++, versione 5.3.0
- SDK per server C#, versione 5.3.0
- Go server SDK, versione 5.3.0

- Server SDK per Unreal, versione 5.3.0
- Server SDK per Unity, versione 5.3.0

Versioni aggiornate del plugin:

- Plugin per Unreal, versione 3.0.0
- Plugin per Unity, versione 3.1.0

Le nuove versioni del server SDK per C++, C#, Go, Unreal e Unity, nonché le nuove versioni dei plugin per Unreal Engine e Unity, sono ora open source. Sono tutte disponibili nell'organizzazione. Amazon GameLift Servers GitHub Per informazioni dettagliate sugli aggiornamenti, consulta le note di rilascio e i readme in ogni repository.

Aggiornamenti chiave dell'SDK del server:

- Convalida lato client e risposte agli errori migliorate in tutti i server. SDKs
- La funzione di OnProcessTerminatecallback ora ha una logica predefinita per terminare il processo del server di gioco.
- La funzione InitSDK() ora utilizza un token di idempotenza per supportare più tentativi.
- Il OnUpdateGameSessioncallback può ora passare dati per. autoBackfillMode

Aggiornamenti chiave del plugin:

- Il plugin per Unreal Engine ora ha un processo di installazione e configurazione più semplificato, con maggiore automazione e meno prerequisiti (CMakeOpenSSL e la catena di strumenti di compilazione incrociata Unreal).
- Il plug-in per Unreal Engine ha migliorato l'esperienza dell'interfaccia utente per il EC2 flusso di lavoro gestito, incluso il supporto per gli spazi nei percorsi di creazione del client di gioco e del server. Inoltre, ora puoi aggiungere argomenti della riga di comando quando avvii un client di gioco dall'Editor.
- Il plugin per Unreal Engine ora supporta le build del server ARM. UE5

Ulteriori informazioni:

- Amazon GameLift Serversplugin per Unreal Engine, Amazon GameLift Servers Guida per gli sviluppatori

- Amazon GameLift Serversplugin per Unity (server SDK 5.x), Guida Amazon GameLift Servers per gli sviluppatori
- Server SDK 5.x per Amazon GameLift Servers, Guida Amazon GameLift Servers per gli sviluppatori
- Integrazione Amazon GameLift Servers in un progetto Unreal Engine, Guida Amazon GameLift Servers per gli sviluppatori
- Integrazione Amazon GameLift Servers in un progetto Unity, Guida Amazon GameLift Servers per gli sviluppatori

24 aprile 2025: il server C# SDK 5 per Amazon GameLift Servers ora supporta .NET 8

Versioni SDK aggiornate:

- C# Server SDK, versione 5.2.1

Per gli sviluppatori di giochi che utilizzano C#, ora puoi utilizzare.NET 8 come framework di destinazione per i tuoi progetti. Amazon GameLift Servers Con .NET 8, puoi sfruttare i miglioramenti delle prestazioni, tra cui una migliore compilazione just-in-time (JIT), l'ottimizzazione dell'utilizzo della memoria e tempi di avvio più rapidi. Se attualmente utilizzi .NET 6, ti consigliamo di pianificare una migrazione a .NET 8, incluso l'aggiornamento dell'SDK del server C# alla versione più recente. Microsoft ha annunciato il supporto di.NET 8, con patch di sicurezza e aggiornamenti tecnici, fino a novembre 2026.

Scarica la versione più recente dell'SDK del server C# per Amazon GameLift Servers

Ulteriori informazioni:

- Ottieni strumenti Amazon GameLift Servers di sviluppo, Amazon GameLift Servers Guida per gli sviluppatori

27 marzo 2025: Amazon GameLift Servers amplia il supporto per le famiglie di istanze di nuova generazione EC2

Versioni SDK aggiornate:

- AWS SDK 1.11.535

Ora puoi ottimizzare l'hosting del tuo server di gioco Amazon GameLift Servers scegliendo tra una selezione più ampia di istanze Amazon appartenenti alle famiglie di EC2 istanze di quinta e ottava generazione. Ogni nuova EC2 generazione offre miglioramenti in termini di EC2 elaborazione, memoria e rete, con le istanze di ottava generazione che offrono istanze all'avanguardia basate su Graviton4 e Intel Xeon. AWS Le istanze di nuova generazione sono disponibili nelle seguenti famiglie di istanze:

- Uso generico (serie M)
- Ottimizzato per il calcolo (serie C)
- Memoria ottimizzata (serie R)

È inoltre possibile scegliere varianti che offrono storage locale (d), rete avanzata (n) e architetture di processore specifiche (). Intel/AMD/Graviton – i/a/g Le istanze di nuova generazione sono disponibili in tutte Regioni AWS le aree supportate daAmazon GameLift Servers, ad eccezione delle regioni della Cina AWS . Per ulteriori dettagli, consulta Amazon GameLift Serverssedi di assistenza.

Usa questi nuovi tipi di istanze con EC2 flotte Amazon GameLift Servers gestite e flotte di container gestite. Quando passi l'hosting di gioco esistente a un nuovo tipo di istanza (stessa architettura), è sufficiente implementare nuove flotte con tutte le impostazioni di configurazione invariate tranne il tipo di istanza.

Ulteriori informazioni:

- Amazon GameLift ServersPrezzi delle istanze
- CreateFleetAmazon GameLift ServersRiferimento alle API
- CreateContainerFleet, Riferimento Amazon GameLift Servers API

11 marzo 2025: Amazon GameLift Servers lancia il nuovo Game Server Wrapper per un rapido onboarding

Il nuovo game server wrapper riduce Amazon GameLift Servers significativamente il tempo necessario per ospitare un server di gioco su cui è ospitato. Amazon GameLift Servers Senza dover modificare il codice, puoi utilizzare il wrapper per aggiungere funzionalità di base per la gestione delle sessioni di gioco al gioco e distribuirlo su una flotta Amazon GameLift Servers Anywhere, una flotta gestita o una flotta di EC2 container gestiti. Questo strumento è ideale per effettuare una valutazione pratica delle Amazon GameLift Servers funzionalità, utilizzando il proprio server di gioco o uno di

un gioco di esempio. È utile anche per implementare rapidamente iterazioni di server di gioco, ad esempio per prototipazione o test rapidi.

Grazie alle funzionalità di base per la gestione delle sessioni di gioco, il server di gioco può inizializzare una connessione con il Amazon GameLift Servers servizio, rispondere alle richieste di avvio e interruzione delle sessioni di gioco e spegnersi al termine di una sessione di gioco.

Scarica il wrapper del server di gioco da. GitHub Consulta il readme su come installare e utilizzare il GitHub wrapper con i tuoi progetti di gioco.

6 marzo 2025: Amazon GameLift è ora Amazon GameLift Servers e Amazon Streams GameLift

Amazon GameLift è un servizio completamente gestito dedicato ad aiutare gli sviluppatori a creare, scalare e fornire i giochi più impegnativi al mondo. Con la versione di disponibilità generale di Amazon GameLift Streams, Amazon offre GameLift ora sia server di gioco ad alta scalabilità che funzionalità di streaming di gioco fluide.

Amazon GameLift Serversoffre agli sviluppatori di giochi la possibilità di implementare, gestire e scalare server di gioco dedicati. Puoi implementare server di gioco ad alte prestazioni nel cloud in pochi minuti e scalare verso l'alto e verso il basso per soddisfare le esigenze dei giocatori. Basato su un ambiente informatico AWS collaudato, Amazon GameLift Servers supporta 100 milioni di giocatori simultanei in una singola partita, 100.000 aggiunte di giocatori al secondo e 9.000 nuove istanze di calcolo al minuto. Inoltre, grazie alla sicurezza di livello aziendale, al matchmaking per il pubblico più numeroso e alla pay-as-you-go flessibilità, ti aiuta a iniziare sia che tu stia lavorando a una nuova idea di gioco sia che tu stia gestendo una partita con milioni di giocatori.

Amazon GameLift Streams aiuta gli sviluppatori di giochi a offrire esperienze di streaming di gioco con una risoluzione fino a 1080p e 60 frames-per-second (fps) senza ritardi percepibili su dispositivi tra cui iOS, Android, FireOS e per i giocatori. PCs Utilizzando un'unica AWS offerta, gli editori possono distribuire i propri contenuti di gioco in pochi minuti, senza modifiche, su istanze GPU basate su cloud completamente gestite e distribuirli tramite AWS Network Backbone direttamente a qualsiasi dispositivo del giocatore finale dotato di browser web. I giocatori possono iniziare a giocare in pochi secondi, senza attendere il download o l'installazione e offre un'esperienza di gioco quasi indistinguibile dal gioco in locale su un PC o una console di gioco.

14 gennaio 2025: Amazon GameLift Servers rilascia il supporto esteso per la prioritizzazione del posizionamento delle sessioni di gioco

Versioni SDK aggiornate:

- AWS SDK 1.11.485

In risposta al feedback dei clienti, stiamo rilasciando una nuova funzionalità che consente di assegnare priorità alle posizioni per le richieste di posizionamento delle singole sessioni di gioco. Per le code configurate per dare priorità al posizionamento in base alla località, ora puoi fornire un elenco personalizzato di posizioni prioritarie con ogni richiesta di posizionamento.

Questa nuova funzionalità consente ai clienti di modificare dinamicamente le priorità di localizzazione per ogni richiesta di collocamento in base alle esigenze. La flessibilità aggiuntiva consente di rispondere meglio alle mutevoli condizioni, come la posizione dei giocatori, il carico della flotta o lo stato del server. Può anche supportare i clienti che desiderano personalizzare ulteriormente il modo in cui vengono selezionate le sedi di collocamento.

Ulteriori informazioni:

- Dai priorità al posizionamento delle sessioni di gioco, Amazon GameLift Servers Guida per gli sviluppatori
- StartGameSessionPlacement, Amazon GameLift Servers Riferimento alle API

2 gennaio 2025: Amazon GameLift Servers rilascia un nuovo supporto per l'interruzione delle sessioni di gioco

Versioni SDK aggiornate:

- AWS SDK 1.11.477

In risposta al feedback dei clienti, stiamo rilasciando nuove funzionalità che consentono di terminare più facilmente le singole sessioni di gioco. Con questa versione, ora puoi terminare una sessione di gioco direttamente nella Amazon GameLift Servers console o utilizzando l'SDK AWS CLI o AWS per. Amazon GameLift Servers

Questa nuova funzionalità risponde alla necessità di risolvere sessioni di gioco che rimangono attive ma in cattivo stato, impedendo così alle risorse di calcolo di ospitare nuove sessioni di gioco. In precedenza, i clienti dovevano accedere in remoto al computer per terminare manualmente una sessione di gioco.

Hai due metodi di terminazione tra cui scegliere. Il primo metodo tenta di terminare correttamente una sessione di gioco utilizzando la sequenza di spegnimento personalizzata, che potrebbe includere

azioni per avvisare i giocatori e risolvere i dati di gioco. Il secondo metodo impone l'interruzione del processo del server, che interrompe immediatamente la sessione di gioco. Questo secondo metodo assicura che la sessione di gioco termini anche quando il processo del server non risponde.

Ulteriori informazioni:

- Chiudi una sessione di gioco utilizzando la Amazon GameLift Servers console, Guida per Amazon GameLift Serversgli sviluppatori
- TerminateGameSession, Riferimento Amazon GameLift Servers API

19 dicembre 2024: Amazon GameLift Servers rilascia il supporto per i plugin del motore di gioco per Managed Containers

Versioni aggiornate del plugin:

Amazon GameLift Serversplugin per Unreal Engine, versione 2.0.0

- Aggiornato per supportare il server C++ SDK 5.2.0 con supporto per contenitori gestiti.
- È stato aggiunto il supporto per Unreal Engine 5.4 e 5.5.

Amazon GameLift Serversplugin per Unity, versione 3.0.0

- Aggiornato per supportare il server C++ SDK 5.2.0 con supporto per contenitori gestiti.
- Support per Unity 2021.3 LTS e 2022.3 LTS per Windows e Mac OS.

Il Amazon GameLift Servers plug-in per i motori di gioco Unreal e Unity fornisce strumenti e flussi di lavoro che semplificano i passaggi per avviare e utilizzare un gioco. Amazon GameLift Servers Amazon GameLift Serversè un servizio di hosting cloud completamente gestito che gli sviluppatori di giochi possono utilizzare per gestire e scalare server di gioco dedicati per giochi multiplayer basati su sessioni.

Le ultime versioni dei plugin offrono i seguenti miglioramenti:

- Flusso di lavoro guidato per l'hosting con Managed Containers. Questo flusso di lavoro illustra i passaggi per configurare un'immagine del contenitore con il software del server di gioco e implementare una soluzione di hosting basata su cloud per il server di gioco. Il flusso di lavoro offre due diversi scenari di implementazione: una distribuzione semplice e una distribuzione più

completa con una coda per il posizionamento delle sessioni di gioco e un matchmaker. FlexMatch Ogni scenario genera flotte di Amazon GameLift Servers container e risorse di supporto. AWS

- Processo migliorato per la configurazione dei profili AWS utente e la gestione delle credenziali di AWS accesso per l'uso dei plug-in. Puoi gestire più profili per lavorare con AWS account, utenti di account e aree geografiche diversi.
- Funzionalità aggiuntive per aggiornare le flotte di container esistenti. Puoi distribuire nuove immagini di container (ad esempio per gli aggiornamenti delle versioni dei server di gioco) e modificare le impostazioni di configurazione della flotta senza dover ricominciare dall'inizio.
- Flussi di lavoro migliorati per l'hosting con flotte Amazon GameLift Servers Anywhere e Managed. EC2 I miglioramenti basati sul feedback dei clienti includono una guida migliore con suggerimenti e link a risorse utili.

Gli scenari di implementazione per i container gestiti e EC2 le soluzioni gestite utilizzano AWS CloudFormation modelli per creare e distribuire le AWS risorse per ogni scenario. Questi modelli sono inclusi nel download del Amazon GameLift Servers plug-in e sono modificabili. Puoi usarli così come sono o modificarli per il tuo gioco.

Ulteriori informazioni:

- Plugin per Unreal: distribuisci il tuo gioco su una flotta di container gestita, Amazon GameLift Servers Guida per sviluppatori
- Plugin per Unity: distribuisci il gioco su una flotta di container gestita, Guida per sviluppatori Amazon GameLift Servers
- Scarica il plugin per Unreal da GitHub
- Scarica il plugin per Unity da GitHub
- Hosting di giochi con Amazon GameLift Servers
- Forum Amazon GameLift Servers

12 novembre 2024: Amazon GameLift Servers lancia l'hosting di giochi multiplayer con Managed Containers

Versioni SDK aggiornate:

- AWS SDK 1.11.445
- Server SDK, versione 5.2.0 (tutte le lingue)

Amazon GameLift Serversrilascia per la disponibilità generale una nuova soluzione di hosting per carichi di lavoro su server di gioco containerizzati. Con questa versione, gli sviluppatori di giochi possono ora sfruttare i vantaggi della containerizzazione, tra cui ambienti coerenti e sicuri, un processo di distribuzione semplificato e un utilizzo ottimizzato delle risorse.

Le flotte di container gestite utilizzano EC2 istanze Amazon gestite per tuo conto e Amazon GameLift Servers in base alle tue configurazioni. Puoi creare un'architettura di container personalizzata per il tuo gioco e fornire immagini di container archiviandole in un repository Amazon Elastic Container Registry (Amazon ECR). Le flotte di container sono disponibili solo per i server di gioco basati su Linux. I server di gioco devono essere integrati con Server SDK 5.2.0 o versione successiva.

Con le flotte di container gestite, ottieni gli stessi vantaggi delle flotte gestite. EC2 Ciò include il supporto per i tipi di istanze On-Demand e Spot, la scalabilità intelligente della capacità, il posizionamento delle sessioni di gioco con code e il matchmaking. Inoltre, ottieni le stesse metriche degli altri tipi di flotta, oltre ad alcune nuove metriche per i container. Altre caratteristiche per le flotte di container includono:

- Allineamento con l'esperienza serverless per carichi di lavoro containerizzati. Esegui un processo del server di gioco per container e raccogli molti container in ogni istanza del parco macchine per un utilizzo ottimale delle risorse. Se preferisci avere contenitori con più processi del server di gioco, puoi utilizzare l'Amazon GameLift Serversagente per la gestione automatizzata degli host.
- Creazione semplificata della flotta. Le flotte di container sono progettate per richiedere impostazioni di configurazione di implementazione minime, con valori suggested/default ragionevoli. È possibile implementare rapidamente una flotta funzionante e quindi personalizzare le impostazioni individuali in base alle esigenze.

- Strumenti di controllo delle versioni per l'architettura dei container. Ora puoi aggiornare la definizione di un gruppo di container (che è simile a un «task» di container), mantenere più versioni e specificare quale versione distribuire in una flotta.
- Strumenti di aggiornamento della flotta. Con le flotte di container, non è più necessario creare una nuova flotta quando si desidera rilasciare un aggiornamento della versione del server di gioco. Invece, ora puoi aggiornare l'immagine del container e distribuire gli aggiornamenti alle flotte esistenti.

Puoi creare flotte di Amazon GameLift Servers container Regione AWS ovunque Amazon GameLift Servers supporti flotte con più sedi e distribuire istanze di flotte di container in qualsiasi posizione

remota supportata. Per ulteriori dettagli, consulta Amazon GameLift Serverssedi di assistenza. I container gestiti non sono attualmente disponibili nelle regioni AWS della Cina.

Ulteriori informazioni:

- Post del blog: Sfrutta i container completamente gestiti per ospitare partite multiplayer su scala globale su Amazon GameLift Servers
- Contenitori gestitipanoramica, guida per Amazon GameLift Serversgli sviluppatori
- Come funzionano i contenitori in Amazon GameLift Servers, Guida Amazon GameLift Servers per gli sviluppatori
- Roadmap di sviluppo per l'hosting con contenitori Amazon GameLift Servers gestiti, Guida Amazon GameLift Servers per gli sviluppatori
- CreateContainerFleet, Riferimento Amazon GameLift Servers API

19 settembre 2024: Amazon GameLift Servers rilascia l'aggiornamento dell'SDK del server C++ e dei plugin per Unreal Engine

Versioni SDK aggiornate:

C++ Server SDK, versione 5.1.3

- Nuove funzionalità di registrazione. Ora puoi accedere ai registri delle richieste SDK.
- Migliore affidabilità della trasmissione dei messaggi SDK. L'SDK ora utilizza meccanismi di riconnessione più robusti per il ripristino in caso di interruzioni di rete o cadute casuali dei messaggi.

Versioni aggiornate del plugin:

Amazon GameLift Serversplugin per Unreal Engine, versione 1.1.2

- Aggiornato per supportare l'ultima versione del server C++ SDK 5.1.3.
- Nel Amazon GameLift Servers plugin per Unreal Engine, quando cerchi un eseguibile di build del server per una flotta, ora hai la possibilità di sfogliare Tutti i file.

Plugin SDK C++ Server per Unreal, versione 5.1.2

- Aggiornato per supportare l'ultima versione del server C++ SDK 5.1.3.

Ulteriori informazioni:

- Integrazione dei giochi con il Amazon GameLift Servers plug-in per Unreal Engine, Guida per gli sviluppatori Amazon GameLift Servers
- Amazon GameLift Serversdownload di plugin e SDK

5 settembre 2024: Amazon GameLift Servers migliora l'osservabilità del processo di creazione della flotta

Sulla base del feedback dei clienti, abbiamo chiarito il Amazon GameLift Servers flusso di lavoro per creare una EC2 flotta gestita e prepararla per ospitare sessioni di gioco. Miglioramenti:

- Abbiamo fornito descrizioni più specifiche e accurate di ogni fase del processo di creazione della flotta. Questa visibilità migliorata semplifica l'individuazione e la risoluzione dei problemi più rapidamente.
- Le fasi di creazione e attivazione separano meglio le attività di distribuzione delle istanze (creazione) dalle attività per avviare i processi del server di gioco e connettersi al Amazon GameLift Servers servizio (attivazione). Questa modifica semplifica il riconoscimento della probabile causa dei problemi. Inoltre, ora puoi connetterti in remoto alle flotte quando sono in fase di attivazione.
- Due nuovi eventi di creazione della flotta comunicano l'esito positivo o negativo degli script di installazione dei server di gioco. Se la build del server di gioco include uno script di installazione, Amazon GameLift Servers tenta di eseguire lo script ed emette uno dei seguenti nuovi eventi:
  - `FLEET_CREATION_COMPLETED_INSTALLER`
  - `FLEET_CREATION_FAILED_INSTALLER`

Ulteriori informazioni:

- Come funziona Amazon GameLift Servers la creazione della flotta, Guida per Amazon GameLift Serversgli sviluppatori
- Debug dei problemi del parco istanze di Amazon GameLift Servers, Guida Amazon GameLift Servers per gli sviluppatori
- Tipo di dati dell'evento, riferimento Amazon GameLift Servers API

15 agosto 2024: Amazon GameLift Servers migliora l'esperienza della console

In base al feedback dei clienti, abbiamo apportato i seguenti aggiornamenti all'esperienza della Amazon GameLift Serversconsole:

- Le tue preferenze di visualizzazione per le pagine vengono ora salvate automaticamente nell' AWS account utente e applicate ogni volta che torni alla pagina. Le preferenze di visualizzazione consentono di scegliere quali informazioni includere in una tabella, ad esempio nella pagina di elenco Fleets. Personalizza le tue preferenze di visualizzazione utilizzando l'  icona nell'angolo in alto a destra di una tabella.
- Il flusso di lavoro Create Fleet per EC2 flotte gestite è stato semplificato per combinare la selezione di sedi del parco veicoli e tipi di istanze. Ti abbiamo semplificato la ricerca del tipo di istanza giusto per la tua flotta, anche quando modifichi le selezioni delle sedi.

Ulteriori informazioni:

- Crea una EC2 flotta Amazon GameLift Servers gestita, Guida per Amazon GameLift Serversgli sviluppatori

25 luglio 2024: Amazon GameLift Servers aggiunge il supporto per la zona locale AWS della Nigeria

Con l'hosting Amazon GameLift Servers gestito, ora puoi distribuire risorse di server di gioco in Nigeria, Africa occidentale ed estendere la portata dei tuoi giochi ai giocatori di tutta l'Africa. Usa AWS Local Zones per posizionare i server di gioco geograficamente più vicini ai tuoi giocatori per ridurre la latenza e migliorare significativamente il gameplay.

Per iniziare subito a ospitare sessioni di gioco in Nigeria, aggiungi la nuova Zona locale della Nigeria come postazione remota a una flotta multisede nuova o esistente. Se il gioco lo utilizza Amazon GameLift ServersFlexMatch, aggiorna le flotte nella coda di matchmaking per includere la nuova zona locale. Con le flotte con più sedi, puoi gestire direttamente la capacità di hosting in ciascuna località.

L'origine della zona locale Regione AWS di Lagos, in Nigeria, è la regione Africa (Città del Capo) `af-south-1` (), Amazon GameLift Servers che funge anche da postazione remota. Il nome della zona locale della Nigeria è. `af-south-1-los-1`

Ulteriori informazioni:

- Amazon GameLift Serverssedi di assistenza, Guida per Amazon GameLift Serversgli sviluppatori
- Aggiorna le posizioni della flotta, Guida Amazon GameLift Servers per gli sviluppatori

02 luglio 2024: Amazon GameLift Servers rilascia un nuovo strumento su console per visualizzare i dati della sessione dei giocatori

La Amazon GameLift Servers console offre ora uno strumento di ricerca della sessione giocatore che consente di recuperare le informazioni sulla sessione del giocatore in base all'ID della sessione di gioco, all'ID della sessione del giocatore o all'ID giocatore. I giochi che utilizzano il FlexMatch matchmaking generano automaticamente sessioni di gioco per ogni giocatore abbinato. Per tutti gli altri giochi, le sessioni dei giocatori sono una funzionalità opzionale.

Puoi trovare lo strumento di ricerca delle sessioni dei giocatori nella navigazione principale della Amazon GameLift Servers console. Visualizza le sessioni dei singoli giocatori o confronta i dati tra sessioni con più giocatori. Puoi anche aprire i dati della sessione del giocatore quando visualizzi la pagina dei dettagli di una sessione di gioco.

Ulteriori informazioni:

- Sessioni di gioco e di gioco nella Amazon GameLift Servers console, Guida Amazon GameLift Servers per gli sviluppatori

24 aprile 2024: Amazon GameLift Servers rilascia l'anteprima delle flotte di container

Amazon GameLift Serversoffre ora un'anteprima delle flotte di container, che offrono portabilità, scalabilità, tolleranza agli errori e agilità migliorate.

Nelle flotte di container, EC2 le istanze Amazon ospitano uno o più container. Questi contenitori includono il tuo server di gioco e tutto ciò che richiede, comprese le dipendenze e le configurazioni. Esempi di dipendenze includono pacchetti software SDKs . Dopo aver caricato il container nel tuo Amazon Elastic Container Registry privato, Amazon GameLift Servers popola la tua flotta con il container.

Per funzionare in una flotta di container, il server di gioco deve funzionare in Linux ed essere integrato con Server SDK 5.x. In una flotta di container, hai un controllo preciso delle risorse di hosting in modo da poter ottimizzare il consumo di risorse come unità CPU e memoria. Puoi anche ospitare più server di gioco in un container per ridurre l'uso di risorse.

In una flotta di container ottieni molti degli stessi vantaggi di altri tipi di flotte, come i tipi di istanze On-Demand, la scalabilità (automatica e manuale), le code e il matchmaking. Inoltre, ottieni le stesse metriche degli altri tipi di flotta, oltre ad alcune nuove metriche per i container. Le flotte di container offrono una copertura globale ai giocatori nelle seguenti aree geografiche:

- ap-northeast-1
- ap-northeast-2
- ap-southeast-2
- eu-central-1
- eu-west-1
- us-east-1
- us-west-2

Per raggiungere ancora più regioni e zone locali, crea flotte di container con più sedi.

Ulteriori informazioni:

- Gestione dell'hosting con Amazon GameLift Servers contenitori, Guida per gli sviluppatori Amazon GameLift Servers
- CreateContainerGroupDefinition, Riferimento Amazon GameLift Servers API

13 febbraio 2024: Amazon GameLift Servers lancia miglioramenti e semplifica l'installazione del Amazon GameLift Servers plug-in per SDKs Unreal Engine

Versioni SDK aggiornate:

- Go Server SDK, versione 5.1.0
- C# Server SDK, versione 5.1.2
- C++ Server SDK, versione 5.1.2

Abbiamo apportato i seguenti miglioramenti:

- È stata migliorata l'affidabilità dell'SDK aggiungendo la riconnessione automatica in caso di interruzione della rete.

- [Go] Ora puoi chiamare `InitSDK()` con o senza i parametri del server. I server di gioco che funzionano su EC2 flotte Amazon GameLift Servers gestite leggono i parametri del server direttamente dalle variabili di ambiente. I server di gioco delle flotte Amazon GameLift Servers Anywhere devono effettuare chiamate `InitSDK()` con i parametri del server.

Versioni aggiornate dei plugin:

- Amazon GameLift Serversplugin per Unreal Engine, versione 1.1.0
- Amazon GameLift Serversplugin per Unity, versione 2.1.0
- Plugin C++ Server SDK per Unreal, versione 5.1.1
- Plugin C# Server SDK per Unity, versione 5.1.2

Abbiamo apportato i seguenti miglioramenti:

- [Amazon GameLift Serversplugin per Unreal Engine] Ha aggiornato le istruzioni di installazione e semplificato la confezione. Questo plugin ora include l'ultima versione di C++ Server SDK per Unreal.
- Sono stati aggiornati i plugin per supportare la versione più recente del server SDK per. Amazon GameLift Servers

Ulteriori informazioni:

- Integrazione dei giochi con il Amazon GameLift Servers plug-in per Unreal Engine, Guida per gli sviluppatori Amazon GameLift Servers
- Amazon GameLift Serversdownload di plugin e SDK

14 dicembre 2023: Amazon GameLift Servers aggiunge la possibilità di aggiornare le proprietà di gioco delle sessioni di gioco attive

Sei già stato in grado di impostare le proprietà del gioco durante la creazione di sessioni di gioco e di cercare nelle sessioni di gioco proprietà specifiche. Ora puoi anche aggiungere e aggiornare queste proprietà in una sessione di gioco attiva.

Ad esempio, i tuoi giocatori votano su una mappa su cui vogliono giocare. Il client di gioco chiama `UpdateGameSession` per modificare un `GameProperty` valore in`{"Key": "map",`

`"Value":"jungle"}`. Il gioco implementa quindi la nuova mappa per i giocatori nella sessione di gioco.

Gli amministratori del gioco possono anche recuperare dati utili dalle proprietà del gioco utilizzando l'operazione. `SearchGameSessions` Ad esempio, gli amministratori possono elencare le sessioni di gioco che hanno un `Status` valore `ACTIVE` e questa proprietà di gioco:. `{"Key": "map", "Value":"desert"}`

Ulteriori informazioni:

- the section called “Aggiungi Amazon GameLift Servers a un client di gioco”, Guida per Amazon GameLift Serversgli sviluppatori
- GameProperty, Riferimento Amazon GameLift Servers API
- UpdateGameSession, Riferimento Amazon GameLift Servers API
- SearchGameSessions, Riferimento Amazon GameLift Servers API

21 novembre 2023: Amazon GameLift Servers lancia il supporto per strumenti Infrastructure as Code come Terraform e Pulumi con tecnologia AWS Cloud Control API

Ora puoi gestire l'intero stack di Amazon GameLift Servers risorse utilizzando gli strumenti Infrastructure as Code (IaC). Questi strumenti includono AWS CloudFormation e anche strumenti di terze parti come Terraform e Pulumi. Con questo supporto aggiuntivo, ora puoi concentrarti sullo sviluppo del tuo gioco e sfruttare DevOps le strategie per occuparti della gestione delle risorse, della CI/CD e dell'implementazione per i tuoi clienti.

Ora puoi anche effettuare il provisioning e configurare tutti i tipi di Amazon GameLift Servers risorse utilizzando l'API AWS Cloud Control. Puoi continuare a lavorare con le risorse utilizzando Amazon GameLift Servers APIs o i AWS CloudFormation modelli perAmazon GameLift Servers.

Per i dettagli sulle Amazon GameLift Servers risorse disponibili tramite IAc, vedere il riferimento sul tipo di Amazon GameLift Servers risorsa di riferimento Amazon GameLift Servers sul tipo di risorsa.

Inoltre, ora puoi ridimensionare automaticamente le tue flotte utilizzando i AWS CloudFormation modelli o l'API AWS Cloud Control utilizzando la nuova proprietà Fleet:. `ScalingPolicies`

L'API Cloud Control offre agli sviluppatori un set standard di risorse APIs per creare, leggere, aggiornare, eliminare ed elencare le risorse (CRUDL) su centinaia di AWS servizi e diversi strumenti di terze parti come Terraform e Pulumi.

Ulteriori informazioni:

- AWS CloudFormation
- AWS API Cloud Control
- AWS Fornitore CC Terraform
- Pulumi

16 novembre 2023: Amazon GameLift Servers aggiorna il plugin standalone per Unity

Versioni SDK aggiornate: Amazon GameLift Servers plugin per Unity, versione 2.0.0

Il Amazon GameLift Servers plug-in per Unity fornisce strumenti e flussi di lavoro che semplificano i passaggi per rendere operativo il gioco Unity per l'hosting su cloud. Amazon GameLift Servers Amazon GameLift Serversè un servizio completamente gestito che consente agli sviluppatori di giochi di gestire e scalare server di gioco dedicati per giochi multiplayer basati su sessioni.

Con questa versione, il plug-in per Unity viene aggiornato per utilizzare le Amazon GameLift Servers funzionalità più recenti, tra cui la versione 5.x dell'SDK del server e il supporto per i test locali con Anywhere. Amazon GameLift Servers Il plugin è compatibile con le versioni Unity 2021.3 LTS e 2022.3 LTS.

Le funzionalità principali del plug-in includono:

- Flussi di lavoro guidati dell'interfaccia utente nell'editor Unity per i seguenti scenari:
  - Metti alla prova la tua integrazione di gioco Amazon GameLift Servers usando la tua workstation locale come host. Questo flusso di lavoro ti aiuta a configurare una flotta Amazon GameLift Servers Anywhere per il tuo computer locale, avviare istanze del server e del client di gioco, richiedere una sessione di gioco e partecipare al gioco. Amazon GameLift Servers
  - Implementa una soluzione di hosting cloud per il tuo server di gioco integrato con risorse Amazon GameLift Servers gestite EC2 e di supporto AWS . Questo flusso di lavoro ti aiuta a configurare il gioco per l'hosting su cloud e offre tre scenari di implementazione:
    - Distribuisci il server di gioco su una singola flotta.
    - Distribuisci il server di gioco su una serie di flotte Spot a basso costo in più regioni. AWS
    - Distribuisci il server di gioco con un matchmaker. FlexMatch
- Possibilità di configurare profili utente collegati a un utente dell' AWS account e impostare una regione predefinita AWS . Puoi gestire più profili per lavorare in AWS account, utenti di account e aree geografiche diversi.

- Comodità speciali che aiutano a semplificare i processi di Amazon GameLift Servers integrazione e implementazione, tra cui:
  - Ogni soluzione di hosting include AWS risorse di supporto, tra cui un pool di utenti Amazon Cognito che fornisce una convalida unica per giocatore IDs e giocatore. Le soluzioni includono anche un bucket Amazon S3 per lo storage, la notifica degli eventi di Amazon SNS, AWS Lambda funzioni e altre risorse.
  - Per il flusso di lavoro Anywhere, il plug-in automatizza le impostazioni dei parametri del server richieste.
  - Per il EC2 flusso di lavoro di Amazon, ogni soluzione di implementazione fornisce un servizio di backend client integrato che utilizza le funzioni Lambda. Il servizio di backend si colloca tra il client di gioco e il Amazon GameLift Servers servizio e gestisce tutte le chiamate dirette al servizio. Amazon GameLift Servers
- Contenuti per i test di integrazione, tra cui risorse e codice per un semplice gioco multiplayer di esempio per illustrare l'integrazione tra server di gioco e client di gioco.
- Documentazione del plug-in con linee guida dettagliate sull'integrazione e codice di esempio.

Tutti gli scenari di implementazione, inclusi quelli per le EC2 flotte Anywhere e Amazon, utilizzano AWS CloudFormation modelli per descrivere e distribuire le AWS risorse per la soluzione del gioco. Questi modelli sono inclusi nel download del Amazon GameLift Servers plugin. Puoi usarli così come sono o personalizzarli per il tuo gioco.

Ulteriori informazioni:

- Amazon GameLift Serversplugin per Unity (server SDK 5.x), Guida Amazon GameLift Servers per gli sviluppatori
- Scarica il plugin da GitHub
- Informazioni sull'Amazon GameLift Servershosting
- Forum Amazon GameLift Servers

2 novembre 2023: Amazon GameLift Servers aggiunge il supporto per le credenziali condivise

Versioni SDK aggiornate: SDK 1.11.193 AWS

La nuova funzionalità di credenziali Amazon GameLift Servers condivise consente alle applicazioni distribuite su EC2 flotte gestite di interagire con altre risorse. AWS Questo aggiornamento riguarda le

applicazioni raggruppate e distribuite insieme ai file binari dei server di gioco integrati con la versione 5.x o successiva dell'SDK del server. (Gli eseguibili dei server di gioco possono già richiedere le credenziali utilizzando l'azione SDK 5.x del server). `GetFleetRoleCredentials()`

Ad esempio, se desideri distribuire la build del tuo server di gioco con un CloudWatch agente Amazon per raccogliere metriche di EC2 istanza e altri dati, l'agente deve essere autorizzato a interagire con le tue CloudWatch risorse. Per fare ciò, devi prima impostare un ruolo (AWS Identity and Access Management IAM) con le autorizzazioni per utilizzare le CloudWatch risorse, quindi configurare una flotta con il ruolo IAM e le credenziali condivise abilitati. Quando Amazon GameLift Servers distribuisce la build del server di gioco su ogni EC2 istanza, genera un file di credenziali condiviso e lo archivia sull'istanza. Tutte le applicazioni sull'istanza possono utilizzare le credenziali condivise. Amazon GameLift Serversaggiorna automaticamente le credenziali temporanee per tutta la durata dell'istanza.

È possibile abilitare le credenziali condivise quando si crea una EC2 flotta gestita utilizzando i seguenti metodi:

- Nel flusso di lavoro per la creazione del parco Amazon GameLift Servers console.
- Quando si chiama l'operazione API del servizio `CreateFleet` utilizzando il nuovo parametro`InstanceRoleCredentialsProvider`.
- Quando si chiama l'operazione AWS CLI `aws gamelift create-fleet` con il parametro. `instance-role-credentials-provider`

Ulteriori informazioni:

- Comunica con altre AWS risorse del tuo parco veicoli, Guida per gli sviluppatori Amazon GameLift Servers
- CreateFleet InstanceRoleCredentialsProvider, Riferimento Amazon GameLift ServersAPI
- Configura un ruolo di servizio IAM, Guida per Amazon GameLift Servers gli sviluppatori

28 settembre 2023: Amazon GameLift Servers rilascia un nuovo plug-in standalone per Unreal Engine Engine

Versioni SDK aggiornate: Amazon GameLift Servers plugin per la versione 1.0.0 di Unreal Engine

Il Amazon GameLift Servers plug-in per Unreal Engine fornisce strumenti e flussi di lavoro che semplificano i passaggi necessari per avviare e utilizzare un gioco per l'hosting su cloud. Amazon

GameLift Servers Amazon GameLift Serversè un servizio completamente gestito che consente agli sviluppatori di giochi di gestire e scalare server di gioco dedicati per giochi multiplayer basati su sessioni. Il plugin supporta le versioni UE 5.0, 5.1 e 5.2. Le caratteristiche principali includono:

- I flussi di lavoro guidati dell'interfaccia utente nell'editor Unreal] seguono i seguenti percorsi:
  - Metti alla prova l'integrazione del gioco Amazon GameLift Servers usando la tua workstation locale come host. Questo flusso di lavoro ti aiuta a configurare una flotta Amazon GameLift Servers Anywhere per il tuo computer locale, avviare istanze del server e del client di gioco, richiedere una sessione di gioco e ottenere informazioni di connessione per la nuova sessione di gioco. Amazon GameLift Servers
  - Implementa una soluzione di hosting EC2 cloud Amazon per il tuo server di gioco integrato. Questo flusso di lavoro ti aiuta a configurare il gioco per l'hosting su cloud e offre tre diversi scenari di implementazione: distribuzione su una singola flotta, distribuzione su una serie di flotte spot in più regioni o distribuzione su un set di flotte con un matchmaker. FlexMatch La soluzione per ogni scenario di implementazione include risorse e risorse di supporto. Amazon GameLift Servers AWS
- Possibilità di configurare profili utente collegati a un AWS account utente e definire una AWS regione predefinita. È possibile gestire più profili per lavorare in AWS account, utenti di account e aree geografiche diversi.
- Comodità speciali che aiutano a semplificare i processi di Amazon GameLift Servers integrazione e implementazione, tra cui:
  - Ogni soluzione di hosting include AWS risorse di supporto, tra cui un pool di utenti Amazon Cognito di base che fornisce un player unico IDs, un bucket Amazon S3 per lo storage, notifiche di eventi Amazon SNS e funzioni. AWS Lambda
  - Per il flusso di lavoro Anywhere, il plug-in automatizza le impostazioni dei parametri del server richieste utilizzando argomenti della riga di comando.
  - Per il EC2 flusso di lavoro di Amazon, ogni soluzione di implementazione fornisce un servizio di backend client integrato che utilizza le funzioni Lambda. Il servizio di backend riceve le richieste dai client di gioco e le trasmette al servizio. Amazon GameLift Servers
- Contenuti per i test di integrazione, tra cui una mappa di gioco iniziale e due mappe di test con progetti di base ed elementi dell'interfaccia utente.
- Documentazione del plug-in con linee guida dettagliate sull'integrazione e codice di esempio.

Tutti gli scenari di implementazione, inclusi quelli per le EC2 flotte Anywhere e Amazon, utilizzano AWS CloudFormation modelli per descrivere le soluzioni. Il plugin utilizza questi modelli per distribuire

Amazon GameLift Servers risorse per il gioco. Questi modelli sono inclusi nel download del Amazon GameLift Servers plugin e sono modificabili. Puoi usarli così come sono o modificarli per il tuo gioco.

Ulteriori informazioni:

- Amazon GameLift Serversplugin per Unreal Engine, Guida Amazon GameLift Servers per gli sviluppatori
- Scarica il plugin da GitHub
- Informazioni sull'Amazon GameLift Servershosting
- Forum Amazon GameLift Servers

17 agosto 2023: Amazon GameLift Servers offre hosting di server di gioco con processori AWS Graviton

Versioni SDK aggiornate: SDK 1.11.144 AWS

Con ora Amazon GameLift Servers puoi ospitare i tuoi giochi nel cloud utilizzando EC2 istanze con processori Graviton. AWS Progettate AWS con processori basati su ARM64, le istanze Graviton offrono il miglior rapporto prezzo/prestazioni per i carichi di lavoro su cloud EC2, con un miglioramento fino al 40% rispetto alle istanze simili basate su x86. I più recenti processori Graviton3 offrono prestazioni di elaborazione migliori fino al 25% rispetto alle versioni precedenti.

ConAmazon GameLift Servers, ora puoi scegliere tra queste nuove istanze della famiglia Graviton: AWS

- Istanze basate su Graviton2: c6g, c6gn, r6g, m6g, g5g
- Istanze basate su Graviton3: c7g, r7g, m7g

Ulteriori informazioni:

- AWS Processore Graviton: scopri i vantaggi e gli usi pratici delle istanze basate su Graviton. EC2
- Guida introduttiva a Graviton: ottieni una panoramica delle istanze basate su Graviton e approfondimenti su come le applicazioni vengono eseguite su di esse a seconda del sistema operativo, delle lingue e dei tempi di esecuzione.

> **Note**
>
> Le istanze Graviton Arm richiedono un server basato su sistema operativo Linux. Amazon GameLift Servers Server SDK 5.1.1 o versione successiva è richiesto per C++ e C#. Server SDK 5.0 o versione successiva è richiesto per Go. Queste istanze non forniscono out-of-the-box supporto per l'installazione di Mono su Amazon Linux 2023 (AL2023) o Amazon Linux 2 (). AL2

27 luglio 2023: Amazon GameLift Servers rilascia il server SDK 5.1.0 con supporto aggiuntivo per lo sviluppo di Unity

Versioni SDK aggiornate: Server SDK for C++, C#/Unity, Unreal 5.1.0

La versione più recente del Amazon GameLift Servers server SDK fornisce aggiornamenti per C++, C# e il plug-in Unreal e un nuovo plug-in da utilizzare con il motore di gioco Unity. Gli sviluppatori di giochi integrano l'SDK del Amazon GameLift Servers server nei server di gioco su cui distribuiscono per l'hosting. Amazon GameLift Servers

L'ultima versione dell'SDK del server contiene i seguenti aggiornamenti, che includono una serie di richieste dei clienti:

- Scarica pacchetti SDK specifici per lingua: il sito di Amazon GameLift Servers download aggiornato contiene pacchetti SDK per ogni lingua. Puoi scaricare le versioni correnti o precedenti.
- Nuovo plug-in SDK per server C# per Unity: il nuovo pacchetto SDK server per Unity contiene librerie C# integrate che puoi installare utilizzando il gestore di pacchetti in Unity Editor (vedi la nuova guida all'integrazione di Unity). Queste librerie includono le dipendenze richieste tramite. UnityNuGet Puoi utilizzare questo plugin con Unity 2020.3 LTS, 2021.3 LTS e 2022.3 LTS per Windows e Mac OS. Supporta i profili.NET Framework e .NET Standard di Unity, con .NET Standard 2.1 e .NET 4.x.
- Soluzione .NET consolidata per C#: il server SDK per C# ora supporta .NET Framework 4.6.2 (aggiornato dalla versione 4.6.1) e.NET 6.0 in un'unica soluzione. .NET Standard 2.1 è disponibile con le librerie create da Unity.
- Aggiornamenti Server SDK 5.1.0
  - [C++, C#, Unreal] Ora puoi chiamare `InitSDK()` con o senza i parametri del server. I server di gioco che funzionano su EC2 flotte Amazon GameLift Servers gestite leggono i parametri del

server direttamente dalle variabili di ambiente. I server di gioco delle flotte Amazon GameLift Servers Anywhere devono effettuare chiamate `InitSDK()` con i parametri del server.

- [C++, C#, Unreal] Le chiamate Server SDK hanno migliorato la messaggistica di errore.
- [C++ SDK] Per migliorare i tempi di compilazione di Server SDK, il flag di compilazione è disabilitato per impostazione predefinita. `-DRUN_CLANG_FORMAT` Puoi abilitarlo con. `-DRUN_CLANG_FORMAT=1`
- [C++ SDK] Quando si creano le librerie senza le librerie standard (`-DGAMELIFT_USE_STD=0`), `InitSDK()` non utilizza più i tipi di `std::` dati.

- Documentazione estesa del server SDK 5.x
  - Guide di riferimento SDK del server aggiornate per C++, C#/Unity e Unreal, inclusa una copertura estesa di tutti i tipi di dati.
    - Server C# SDK 5.x per -- Azioni Amazon GameLift Servers
    - Server C++ SDK 5.x per -- Azioni Amazon GameLift Servers
    - Server C++ (Unreal) SDK 5.x per -- Azioni Amazon GameLift Servers
  - Nuove versioni delle guide all'integrazione dell'SDK 5 del server per i plugin Unity e Unreal
    - Integrazione Amazon GameLift Servers in un progetto Unity
    - Integrazione Amazon GameLift Servers in un progetto Unreal Engine
- Aggiornamenti della documentazione aggiuntivi
  - Documentazione rivista per le operazioni delle API di Amazon GameLift Servers servizio GetComputeAccesse GetInstanceAccessper chiarire le procedure di accesso remoto in base alla versione SDK del Amazon GameLift Servers server in uso.
  - Descrizioni riviste per GameSessionPlacementdocumentare come le informazioni sulla sessione di gioco siano transitorie quando un posizionamento è in stato «in sospeso».

13 luglio 2023: Amazon GameLift Servers aggiunge le metriche hardware della flotta

Ora puoi tenere traccia delle metriche delle prestazioni hardware per le tue flotte Amazon GameLift Servers gestite EC2. Le metriche includono le metriche delle EC2 istanze per l'utilizzo della CPU, il volume del traffico di rete e l'attività del disco. read/write InfattiAmazon GameLift Servers, queste metriche descrivono tutte le istanze attive in una flotta. Puoi visualizzare queste metriche hardware della flotta utilizzando una CloudWatch dashboard di Amazon nel AWS Management Console. Puoi anche visualizzarli nella Amazon GameLift Servers console nei dettagli del parco veicoli.

Ulteriori informazioni:

- Monitora Amazon GameLift Servers con Amazon CloudWatch(Metriche per le flotte), Amazon GameLift Servers Guida per gli sviluppatori

29 giugno 2023: Amazon GameLift Servers avvia il supporto per Amazon Linux 2023

Versioni SDK aggiornate: SDK 1.11.111 AWS

Amazon GameLift Serversi clienti possono ora utilizzare il sistema operativo Amazon Linux 2023 per ospitare i propri server di gioco. AL2023 offre diversi miglioramenti rispetto all' AL2 inclusione della sicurezza. Questo sistema operativo è disponibile in tutte le Regioni AWS regioni ad eccezione delle regioni della Cina.

I clienti possono utilizzare i più recenti sistemi operativi Linux e continuare a ricevere aggiornamenti di sicurezza critici quando il supporto per Amazon Linux (AL1) terminerà a dicembre 2023. Il supporto per Amazon Linux 2 continuerà fino al 30 giugno 2025.

Ulteriori informazioni:

- Amazon GameLift ServersDomande frequenti su Linux Server
- Confronto tra Amazon Linux 2 e Amazon Linux 2023
- Amazon GameLift ServersLink di riferimento API:
  - AWS Azione SDK `CreateBuild`
  - Comando CLI `upload-build`
  - Comando CLI `create-build`

25 maggio 2023: Amazon GameLift ServersFleetIQ aggiunge un filtro per escludere i posizionamenti delle sessioni di gioco sulle istanze che si esauriscono

Versioni SDK aggiornate: SDK 1.11.87 AWS

Se lo utilizzi Amazon GameLift Servers FleetIQ per l'hosting di giochi, ora puoi impedire il posizionamento delle sessioni di gioco su istanze attualmente in fase di esaurimento. Le istanze in fase di esaurimento sono contrassegnate per essere chiuse, ma possono comunque essere selezionate per ospitare nuove sessioni di gioco se non sono disponibili altre risorse di hosting. Con questa nuova funzionalità, puoi escludere completamente l'uso di istanze drenanti.

Usa questa funzione quando chiami `ClaimGameServer` per trovare i server di gioco disponibili. Aggiungi il nuovo `FilterOption` parametro e imposta gli stati delle istanze consentiti solo su ACTIVE. In risposta, Amazon GameLift Servers FleetIQ esamina solo le istanze attive durante la ricerca e la rivendicazione di un server di gioco disponibile.

Ulteriori informazioni:

- ClaimGameServer nel Documentazione di riferimento API di Amazon GameLift Servers
- Come FleetIQ funziona nella Guida per gli sviluppatori Amazon GameLift Servers FleetIQ

16 maggio 2023: Amazon GameLift Servers supporta l'etichettatura dell'allocazione dei costi per le flotte

Amazon GameLift Serversi clienti possono ora utilizzare i tag di allocazione AWS Billing dei costi per organizzare i costi di hosting dei giochi. Puoi assegnare tag di allocazione dei costi alle singole risorse del Amazon GameLift Servers EC2 parco veicoli per tenere traccia del contributo delle flotte ai costi complessivi di hosting.

Ulteriori informazioni:

- Gestisci i tuoi Amazon GameLift Servers costi di hosting
- Utilizzo AWS dei tag di allocazione dei costi, Guida per l'utente AWS Billing

20 aprile 2023: Amazon GameLift Servers avvia il supporto per Windows Server 2016

Versioni SDK aggiornate: SDK 1.11.63 AWS

Amazon GameLift Serversi clienti possono ora utilizzare il sistema operativo Windows Server 2016 per ospitare i propri server di gioco. Questo sistema operativo è disponibile in tutte le versioni Regioni AWS. I clienti possono utilizzare il nuovo sistema operativo Windows e continuare a ricevere aggiornamenti di sicurezza critici poiché Microsoft terminerà il supporto per Windows Server 2012 a ottobre 2023.

A partire da oggi, i nuovi clienti che richiedono un ambiente di runtime Windows devono specificare Windows Server 2016 quando creano nuove build di server di gioco per l'hosting. I clienti esistenti possono continuare a creare nuove build e flotte con Windows Server 2012, ma devono completare la migrazione con Windows Server 2016 prima della data di fine del supporto di Microsoft, il 10 ottobre 2023.

Questo aggiornamento include le seguenti modifiche al servizio:

- Quando crei una build di server di gioco utilizzando i comandi Amazon GameLift Servers SDK o CLI, ora devi impostare esplicitamente il sistema operativo. Non esiste più un valore predefinito. Per distribuire il tuo server di gioco su Windows Server 2016, usa il valore`WINDOWS_2016`.
- Quando crei una build di server di gioco utilizzando la Amazon GameLift Servers console, devi selezionare un sistema operativo tra i valori disponibili. Se sei un cliente esistente con flotte Windows Server 2012 attive, puoi scegliere una delle due opzioni `WINDOWS_2012` oppure`WINDOWS_2016`.

Ulteriori informazioni:

- Amazon GameLift ServersLink di riferimento API:
  - Comando CLI `upload-build`
  - Comando CLI `create-build`
  - AWS Azione SDK `CreateBuild`
- Amazon GameLift ServersDomande frequenti per Windows 2012

13 aprile 2023: Amazon GameLift Servers lancia il server SDK 5.x per Unreal

Versioni SDK aggiornate: Server SDK 5.0.0 per Unreal

L'ultima versione del plug-in Amazon GameLift Servers leggero per Unreal Engine è ora basata sul server SDK 5.x. Amazon GameLift Servers Per iniziare a integrare il tuo ambiente Unreal Engine, consulta i Amazon GameLift Servers seguenti link.

Ulteriori informazioni:

- Integrazione Amazon GameLift Servers in un progetto Unreal Engine
- Aggiungi Amazon GameLift Servers al tuo server di gioco
- Server C++ SDK 5.x per -- Azioni Amazon GameLift Servers

14 marzo 2023: Amazon GameLift Servers lancia una nuova esperienza per console

La nuova Amazon GameLift Servers console include i seguenti miglioramenti:

- Navigazione migliorata: il nuovo pannello di navigazione facilita la navigazione tra Amazon GameLift Servers le risorse.
- Amazon GameLift Serverspagina di destinazione: la nuova pagina di destinazione fornisce collegamenti a documentazione utile, mostra una panoramica di Amazon GameLift Servers alto livello e fornisce supporto tramite collegamenti alla documentazione, domande frequenti e. AWS re:Post
- CloudWatch Metriche Amazon migliorate: le Amazon GameLift Servers metriche sono ora disponibili sia nella Amazon GameLift Servers console che nelle dashboard. CloudWatch Questo aggiornamento include anche nuove metriche per prestazioni, utilizzo e sessioni di gioco.

Ulteriori informazioni:

- Tieni traccia dell'hosting di giochi in Amazon GameLift Servers console
- Costruire un matchmaker FlexMatch

14 febbraio 2023: Amazon GameLift Servers ora supporta la crittografia lato server per gli argomenti di Amazon SNS

Server Side Encryption ((SSE)) per gli argomenti SNS crittografa i dati sensibili quando sono inattivi. SSE utilizza le chiavi AWS Key Management Service (AWS KMS) per proteggere il contenuto degli argomenti SNS.

Ulteriori informazioni:

- Imposta la notifica degli eventi per il posizionamento della sessione di gioco
- FlexMatcheventi di matchmaking
- Crittografia dei dati inattivi

9 febbraio 2023: il Amazon GameLift Servers server SDK supporta .NET 6 con C #10

Versioni SDK aggiornate: Server SDK 5.0.0 per .NET 6. Non sono richiesti aggiornamenti SDK.

Se utilizzi la piattaforma di sviluppo Unity Real-Time, continua a utilizzare il Amazon GameLift Servers server SDK 5.0.0 con .NET 4.6. Unity non supporta .NET 6.

Ulteriori informazioni:

- Scarica l'ultima versione dell'SDK Amazon GameLift Servers del server all'Amazon GameLift Serversinizio
- Server C# SDK 5.x per -- Azioni Amazon GameLift Servers

31 gennaio 2023: l'SDK Amazon GameLift Servers del server supporta il linguaggio Go

Versioni SDK aggiornate: Server SDK 5.0.0 per Go

Ulteriori informazioni:

- Scarica la versione più recente dell'SDK del Amazon GameLift Servers server all'inizio Amazon GameLift Servers
- Go server SDK per Amazon GameLift Servers -- Azioni

1 dicembre 2022: Amazon GameLift Servers lancia Amazon GameLift Servers Anywhere e Amazon GameLift Servers Server SDK 5.0

Versioni SDK aggiornate: SDK 1.10.21, Server AWS SDK 5.0.0 per C++ e C#

Amazon GameLift ServersAnywhere utilizza le risorse del server di gioco per ospitare server di gioco. Amazon GameLift Servers Puoi utilizzare Amazon GameLift Servers Anywhere per integrare le tue risorse di elaborazione con l' EC2 elaborazione Amazon GameLift Servers gestita per distribuire i server di gioco su più tipi di elaborazione. Puoi anche usare Amazon GameLift Servers Anywhere per testare in modo iterativo i tuoi server di gioco senza caricare la build per ogni iterazione. Amazon GameLift Servers

Punti salienti:

- Nuovi tipi di flotta e calcolo Amazon GameLift Servers Anywhere
- Amazon GameLift ServersRegistrazione delle risorse di calcolo Anywhere
- Ciclo di iterazione dei test migliorato

Amazon GameLift ServersServer SDK 5.0.0 introduce miglioramenti all'SDK del server esistente e un nuovo tipo di risorsa, compute. Server SDK 5.0.0 supporta Amazon GameLift Servers Anywhere e l'uso delle proprie risorse di elaborazione per l'hosting di server di gioco.

Ulteriori informazioni:

- Server SDK 5.x per Amazon GameLift Servers
- Ubicazione della flotta
- Scegli le risorse di elaborazione per una flotta gestita
- Crea un Amazon GameLift Servers Flotta ovunque

25 agosto 2022: Amazon GameLift Servers lancia il supporto per Local Zones

Versioni SDK aggiornate: SDK 1.9.333 AWS

Amazon GameLift Serversè ora disponibile in otto Local Zones negli Stati Uniti d'America, quindi puoi schierare le tue flotte più vicino ai giocatori. Puoi utilizzare tutte le Amazon GameLift Servers funzionalità gestite con Local Zones aggiungendo le Local Zones alle tue flotte.

Le Local Zones estendono AWS risorse e servizi all'edge del cloud, vicino a grandi centri di popolazione, industria e tecnologia dell'informazione (IT). Ciò significa che è possibile distribuire applicazioni che richiedono una latenza di un millisecondo più vicino agli utenti finali o ai data center locali.

Ulteriori informazioni:

- Amazon GameLift Serverszone locali
- Ubicazione della flotta
- Crea una EC2 flotta Amazon GameLift Servers gestita

28 giugno 2022: Amazon GameLift Servers lancia una nuova esperienza di console opt-in

La nuova Amazon GameLift Servers console include i seguenti miglioramenti:

- Navigazione migliorata: il nuovo pannello di navigazione facilita la navigazione tra Amazon GameLift Servers le risorse.
- Amazon GameLift Serverspagina di destinazione: la nuova pagina di destinazione fornisce collegamenti a documentazione utile, mostra una panoramica di Amazon GameLift Servers alto livello e fornisce supporto tramite collegamenti alla documentazione, domande frequenti e. AWS re:Post

- CloudWatch Metriche Amazon migliorate: le Amazon GameLift Servers metriche sono ora disponibili sia nella Amazon GameLift Servers console che nelle dashboard. CloudWatch Questo aggiornamento include anche nuove metriche per prestazioni, utilizzo e sessioni di gioco.

Ulteriori informazioni:

- Tieni traccia dell'hosting di giochi in Amazon GameLift Servers console
- Costruire un matchmaker FlexMatch

15 febbraio 2022: FlexMatch aggiunge una regola composta e ulteriori miglioramenti

FlexMatchgli utenti ora hanno accesso alle seguenti funzionalità:

- Regola composta: aggiunto il supporto per le regole di matchmaking composte per partite di 40 o meno giocatori. Ora puoi usare istruzioni logiche per creare una regola composta per formare una partita. Senza una regola composta nel set di regole, per creare una corrispondenza, tutte le regole del set di regole devono essere vere. Con le regole composte, è possibile scegliere quali regole applicare utilizzando i seguenti operatori logici: `and`,`or`,`not`, `exor`.
- Selezione flessibile dei team: espressioni delle proprietà di matchmaking aggiornate per supportare la selezione di un sottoinsieme di tutti i team disponibili.
- Elenchi di stringhe più lunghi: è stato aumentato il numero massimo di stringhe da 10 a 100 in un elenco di stringhe di valori degli attributi dei giocatori.

Ulteriori informazioni:

- Amazon GameLift ServersFlexMatchguida per gli sviluppatori:
  - FlexMatchtipi di regole
  - FlexMatchespressioni di proprietà
- AttributeValue: SL

28 ottobre 2021: Amazon GameLift Servers aggiunge il supporto per le flotte multiregionali nella regione Asia Pacifico (Osaka); Amazon GameLift ServersFleetIQ aggiunge il supporto per i processori Graviton2 AWS

Versioni SDK aggiornate: SDK 1.9.133 AWS

Amazon GameLift Servers è ora disponibile nella regione Asia Pacifico (Osaka-Locale). Gli sviluppatori di giochi possono ora distribuire istanze a Osaka utilizzando una flotta multiregionale. GameLift

Ora puoi utilizzare i server di gioco ospitati da Graviton2, basati sull'architettura del processore basata su ARM, per ottenere maggiori prestazioni a un costo inferiore rispetto alle opzioni di elaborazione equivalenti basate su Intel.

Punti salienti:

- Amazon GameLift Servers è ora disponibile nella regione Asia Pacifico (Osaka-Locale).
- Amazon GameLift ServersFleetIQI gruppi di server di gioco possono ora essere configurati per gestire le famiglie di istanze Graviton2 c6g, m6g e r6g.

Ulteriori informazioni:

- Amazon GameLift ServersFlotta multiregionale
- CreateGameServerGroup
- AWS processore gravitonico

20 settembre 2021: Amazon GameLift Servers rilascia il plugin per Unity

Il Amazon GameLift Servers plugin per Unity versione 1.0.0 contiene librerie e interfaccia utente nativa che facilitano l'accesso alle Amazon GameLift Servers risorse e l'Amazon GameLift Serversintegrazione nel gioco Unity. Puoi utilizzare il Amazon GameLift Servers plug-in per Unity per accedere Amazon GameLift Servers APIs e distribuire AWS CloudFormation modelli per scenari di gioco comuni. Il plugin include anche un gioco di esempio che funziona con gli scenari di esempio. Puoi usare Amazon GameLift Servers Local per vedere i messaggi trasmessi tra il client di gioco e il server di gioco e scoprire come interagisce un gioco tipico. Amazon GameLift Servers

Il plugin per Unity supporta Unity 2019.4 LTS e 2020.3 LTS.

Punti salienti:

- Crea, esegui e modifica un gioco di esempio con scenari diversi o creane uno tuo.
- Implementa AWS CloudFormation scenari di esempio per scenari di gioco tipici, tra cui solo autenticazione, flotta a regione singola, flotte multiregionali con coda e matchmaker personalizzato, flotte Spot con coda e matchmaker personalizzato e. FlexMatch

Ulteriori informazioni:

- Amazon GameLift ServersIntegrazione dei giochi con il plug-in per Unity

30 giugno 2021: FlexMatch aggiunge la regola BatchDistance

È possibile utilizzare il tipo di regola BatchDistance per specificare una stringa o un attributo numerico, offrendo una serie di vantaggi a ciascun segmento.

Punti salienti:

- Per partite di grandi dimensioni (più di 40 giocatori), invece di bilanciare equamente i giocatori solo per abilità, ora puoi ottenere lo stesso equilibrio in base all'abilità, alle modalità e alle mappe. Assicurati che tutti i partecipanti alla partita appartengano a un gruppo di abilità, raggruppa più attributi numerici come campionato o stile di gioco e raggruppa in base ad attributi di stringa come mappa o modalità di gioco. Puoi anche creare espansioni nel tempo. Ad esempio, puoi creare un'espansione per consentire a un livello di abilità più ampio di accedere alla partita man mano che il giocatore aspetta.

  Per le partite con meno di 40 giocatori, puoi usare una nuova espressione di regole semplificata.

3 giugno 2021: aggiornamenti dell'SDK del client e dell'SDK del server in Amazon GameLift Servers tempo reale

Versioni SDK aggiornate: Realtime Client SDK 1.2.0, Server SDK 3.4.0 per Unreal

Con questo ultimo aggiornamento SDK, ora puoi integrare IL2 CPP nelle tue applicazioni mobili che utilizzano RTS Client SDK e seguire le migliori pratiche con i framework. Ora puoi anche creare il Amazon GameLift Servers Server SDK per Unreal versione 4.26. Questo aggiornamento contiene componenti che si integrano con il server di gioco Windows o Linux, tra cui le versioni C++ e C# di Amazon GameLift Servers Server SDK, Amazon GameLift Servers Local e un plug-in Unreal Engine.

Punti salienti:

- È stato aggiunto il supporto per IL2 CPP nell'RTS Client SDK e per la creazione di librerie native come framework, in modo da poter creare client RTS per i dispositivi mobili più recenti.
- Puoi utilizzarlo per DescribePlayerSessions() ottenere informazioni per una sessione a giocatore singolo, per tutte le sessioni di gioco o per tutte le sessioni di giocatore associate a un ID giocatore singolo.

- È possibile GetInstanceCertificate() utilizzarlo per recuperare la posizione del file di un certificato TLS con codifica PEM associato alla flotta e alle relative istanze.
- Supporto Created Server SDK per Unreal versione 4.26.
- L'SDK C# esistente, versione 4.0.2, è stata verificata la compatibilità con Unity 2020.3. Non sono stati richiesti aggiornamenti SDK.

Ulteriori informazioni:

- Amazon GameLift ServersGuida per gli sviluppatori:
  - DescribePlayerSessions()
  - GetInstanceCertificate()

23 marzo 2021: Amazon GameLift Servers aggiunge notifiche al posizionamento delle sessioni di gioco

Versioni SDK aggiornate: AWS SDK 1.8.168

Ora puoi utilizzare gli eventi per monitorare l'attività di posizionamento delle sessioni di gioco per una coda di sessione di gioco. Crea un argomento Amazon Simple Notification Service (Amazon SNS) per pubblicare notifiche di eventi o configura il monitoraggio degli eventi tramite Events. CloudWatch

Punti salienti:

- Per ogni coda, puoi impostare una stringa di testo personalizzata da includere in tutti i messaggi relativi agli eventi.
- Quando utilizzi un argomento di Amazon SNS, puoi impostare condizioni di accesso aggiuntive che limitano la pubblicazione a code specifiche.

Ulteriori informazioni:

- Amazon GameLift ServersGuida per gli sviluppatori:
  - Imposta la notifica degli eventi per il posizionamento della sessione di gioco (nuovo)
  - Eventi di collocamento delle sessioni di gioco (nuovo)
- Riferimento all'API (AWS SDK)
  - Nuovi parametri di coda delle sessioni di gioco `NotificationTarget` `eCustomEventData`: GameSessionQueue,, CreateGameSessionQueueUpdateGameSessionQueue

- Forum Amazon GameLift Servers

16 marzo 2021: Amazon GameLift Servers aggiunge flotte multiregionali, sei nuove regioni

Versioni SDK aggiornate: SDK 1.8.163 AWS

Amazon GameLift Serversl'hosting gestito è ora disponibile in 21 regioni. AWS Le nuove regioni sono Città del Capo (`af-south-1`), Bahrein (`me-south-1`), Hong Kong (`ap-east-1`), Milano (), Parigi (`eu-south-1`) e Stoccolma (`eu-west-3`). `eu-north-1`

Con la nuova funzionalità di flotte con Amazon GameLift Servers più sedi, ora puoi configurare un'unica flotta per ospitare i tuoi server di gioco in una o tutte le 20 regioni Amazon GameLift Servers supportate (esclusa la regione di Pechino). Questa funzionalità mira a ridurre in modo significativo il lavoro necessario per configurare e mantenere Amazon GameLift Servers le risorse di hosting a livello globale. È possibile creare flotte con più sedi AWS nelle seguenti regioni: `us-east-1` (Virginia settentrionale), `us-west-2` (Oregon), `eu-central-1` (Francoforte), (`eu-west-1`Irlanda), (Sydney), `ap-southeast-2` (Tokyo) e `ap-northeast-1` (Seoul). `ap-northeast-2` In tutte le altre regioni, puoi continuare a configurare flotte con una sola sede, se necessario. Tutte le flotte create prima di questa versione sono flotte con una sola sede. L'utilizzo di flotte con più sedi non influisce sui costi di hosting. Amazon GameLift Serversi prezzi si basano sul tipo, l'ubicazione e il volume delle istanze utilizzate. (Per ulteriori informazioni, consulta la pagina Amazon GameLift Serversdei prezzi). AWS CloudFormation il supporto per flotte con più sedi sarà presto disponibile.

> **Note**
>
> Le flotte con più sedi non sono disponibili nelle regioni della Cina. Amazon GameLift Serversle risorse che risiedono nelle regioni della Cina non possono interagire o essere utilizzate dalle risorse di altre Amazon GameLift Servers regioni.

Punti salienti:

- Con una flotta con più sedi, aggiungi esplicitamente un elenco di località remote. Amazon GameLift Serversdistribuisce istanze dello stesso tipo e configurazione, inclusa la configurazione di build e runtime, nella regione di origine della flotta e in tutte le sedi aggiunte.

- Regola le impostazioni di capacità e la scalabilità per ogni sede in modo indipendente. Le politiche di scalabilità automatica si applicano a un'intera flotta, ma puoi attivarle o disattivarle in base alla località.
- Inizia nuove sessioni di gioco in località specifiche della flotta. Quando utilizzi le code delle sessioni di gioco o il matchmaking per effettuare le sessioni di gioco, ora puoi dare la priorità a dove iniziano le nuove sessioni di gioco in base alla località, al costo di hosting e alla latenza del giocatore.
- Ottieni le metriche di hosting nella Amazon GameLift Servers console, aggregate per tutte le sedi di un parco veicoli o suddivise per ciascuna località del parco veicoli.

Ulteriori informazioni:

- Blog sulla tecnologia dei giochi di Amazon
- Riferimento alle API (AWS SDK)
  - Nuove operazioni di localizzazione della flotta: CreateFleetLocationsDescribeFleetLocationAttributes,, DescribeFleetLocationCapacity, DescribeFleetLocationUtilizationDeleteFleetLocations
  - Operazioni aggiornate della flotta, con nuovo supporto multisede: CreateFleet,, Descrivi UpdateFleetCapacityEC2InstanceLimits,, DescribeInstances, StopFleetActionsStartFleetActions
  - Operazioni di posizionamento delle sessioni di gioco aggiornate, con nuove funzionalità di priorità e filtro: CreateGameSessionQueue,, DescribeGameSessionQueuesUpdateGameSessionQueue
  - Operazioni di creazione di sessioni di gioco aggiornate, con nuovo supporto di localizzazione: CreateGameSession,, DescribeGameSessions, DescribeGameSessionDetailsSearchGameSessions
- Amazon GameLift ServersGuida per gli sviluppatori:
  - Amazon GameLift Serverssedi di assistenza(aggiornato)
  - Personalizza il tuo Amazon GameLift Servers EC2 flotte gestite (nuovo)

    Scalabilità della capacità di hosting di giochi con Amazon GameLift Servers(aggiornato)
  - Personalizza una coda di sessioni di gioco (nuovo)
  - Dettagli della flotta nel Amazon GameLift Servers console(aggiornato)
- Forum Amazon GameLift Servers

9 febbraio 2021: Amazon GameLift Servers estende il supporto per le istanze AMD in modalità autonoma FlexMatch

Versioni SDK aggiornate: SDK 1.8.139 AWS

Questa versione include i seguenti aggiornamenti:

- Amazon GameLift ServersFleetIQI gruppi di server di gioco possono ora essere configurati per gestire le famiglie di istanze AMD C5a, M5a e R5a. I tipi di EC2 istanze Amazon supportati, elencati per GameServerGroup InstanceDefinition, ora includono quanto segue:
  - c5a.large, c5a.xlarge, c5a.2xlarge, c5a.4xlarge, c5a.8xlarge, c5a.12xlarge, c5a.16xlarge, c5a.24xlarge
  - m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
  - r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge, r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge

  Nota: le istanze AMD per non FleetIQ sono attualmente disponibili per l'uso nella AWS regione Cina (Pechino). Vedi Disponibilità delle funzionalità e differenze di implementazione in Cina.
- Amazon GameLift Serversl'hosting di giochi gestito ora supporta le istanze AMD nella regione Cina (Pechino), gestite da Sinnet. Le nuove famiglie di istanze AMD includono M5a e R5a. I tipi di EC2 istanze supportati, elencati per parco istanze InstanceType, ora includono i seguenti:
  - m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
  - r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge, r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge
- Amazon GameLift ServersFlexMatchora può essere utilizzato come soluzione di matchmaking autonoma nella regione della Cina (Pechino), gestita da Sinnet. I clienti possono creare un FlexMatch matchmaker nella regione di Pechino e configurare il parametro su STANDALONE. FlexMatchMode Per ulteriori informazioni in meritoFlexMatch, con hosting Amazon GameLift Servers gestito o con una soluzione non di Amazon GameLift Servers hosting, nella Guida per gli Amazon GameLift Servers FlexMatch sviluppatori.
- Quando configuri le notifiche di eventi per Amazon GameLift ServersFlexMatch, ora puoi designare un argomento Amazon SNS FIFO come destinazione della notifica. Per ulteriori informazioni, consultare:
  - MatchmakingConfiguration NotificationTarget, Riferimento API Amazon GameLift Servers

- Imposta la notifica FlexMatch degli eventi, Guida per Amazon GameLift Servers FlexMatch gli sviluppatori
- Presentazione di Amazon SNS FIFO: First-in-first-out Pub/Sub messaggistica, notizie e blog AWS

22 dicembre 2020: l'SDK Amazon GameLift Servers del server supporta Unreal Engine 4.25 e Unity 2020

Versioni SDK aggiornate: Amazon GameLift Servers Server SDK 4.0.2, versione 3.3.3 del plugin Unreal

L'ultima versione di Amazon GameLift Servers Server SDK contiene i seguenti componenti:

- Il plugin Unreal aggiornato è stato aggiornato per garantire la compatibilità con Unreal Engine 4.25. L'API non è stata modificata.
- L'SDK C# esistente, versione 4.0.2, è stata verificata la compatibilità con Unity 2020. Non sono stati richiesti aggiornamenti SDK.

Scarica la versione più recente di Amazon GameLift Servers Server SDK all'inizio Amazon GameLift Servers.

24 novembre 2020: Amazon GameLift ServersFlexMatch ora disponibile per i giochi ospitati ovunque

Versioni SDK aggiornate: AWS SDK 1.8.95

Amazon GameLift ServersFlexMatchè un servizio di matchmaking personalizzabile per giochi multiplayer. Progettato inizialmente per gli utenti di hosting Amazon GameLift Servers gestito, ora FlexMatch può essere integrato in giochi che utilizzano altri sistemi di hosting peer-to-peer, tra cui elaborazione proprietaria locale e tipi primitivi di cloud computing. I giochi che utilizzano Amazon GameLift Servers FleetIQ per l'hosting di giochi su Amazon ora EC2 possono implementare il matchmaking conFlexMatch.

FlexMatchfornisce un robusto algoritmo di matchmaking e un linguaggio di regole che ti offrono un'ampia libertà di personalizzazione del processo di matchmaking in modo che i giocatori vengano abbinati in base alle caratteristiche chiave del giocatore e alla latenza riportata. Inoltre, FlexMatch offre un flusso di lavoro per le richieste di matchmaking che supporta funzionalità come i party tra

giocatori, l'accettazione dei giocatori e il riempimento delle partite. Se utilizzi l'FlexMatchhosting Amazon GameLift Servers gestito o Amazon GameLift ServersRealtime, il matchmaker lo utilizza automaticamente Amazon GameLift Servers per trovare risorse di hosting e iniziare una nuova sessione di gioco per le partite appena formate. Quando lo utilizzi FlexMatch come servizio autonomo, il matchmaker invia i risultati delle partite al tuo gioco, che può quindi iniziare una nuova sessione di gioco utilizzando la tua soluzione di hosting.

Le operazioni API per FlexMatch fanno parte dell'API del Amazon GameLift Servers servizio, che è inclusa nell' AWS SDK e in (). AWS Command Line Interface AWS CLI Questa versione include questi aggiornamenti per supportare il matchmaking autonomo:

- La risorsa API `MatchmakingConfiguration` presenta le seguenti modifiche:
  - Nuova proprietà, `FlexMatchMode` indica se il matchmaker viene utilizzato con hosting Amazon GameLift Servers gestito o come matchmaking autonomo.
  - La proprietà non `GameSessionQueueArns` è richiesta quando `FlexMatchMode` è impostata su standalone.
  - Queste proprietà non vengono utilizzate con il matchmaking autonomo:`AdditionalPlayerCount`,,,`BackfillMode`. `GameProperties` `GameSessionData`
- La funzione di riempimento automatico non è disponibile con il matchmaking autonomo.

24 novembre 2020: le istanze AMD sono ora disponibili su Amazon GameLift Servers

Versioni SDK aggiornate: SDK 1.8.95 AWS

L'elenco dei tipi di EC2 istanze Amazon Amazon GameLift Servers attualmente supportati include tre nuove famiglie di istanze: C5a, M5a e R5a. Queste famiglie sono costituite da istanze AMD ottimizzate per il calcolo alimentate da processori AMD EPYC che funzionano a frequenze fino a 3.3. GHz. Le istanze AMD sono compatibili con la tecnologia x86; i giochi attualmente in esecuzione Amazon GameLift Servers possono essere distribuiti su tipi di istanze AMD senza modifiche. Le nuove istanze sono disponibili nelle seguenti AWS regioni: Stati Uniti orientali (Virginia settentrionale e Ohio), Stati Uniti occidentali (Oregon e California settentrionale), Canada centrale (Montreal), Sud America (San Paolo), UE centrale (Francoforte), UE occidentale (Londra e Irlanda), Asia Pacifico meridionale (Mumbai), Asia Pacifico nord-orientale (Seoul e Tokyo) e Asia Pacifico sud-orientale (Singapore e Sydney).

Le nuove istanze AMD includono:

- c5a.large, c5a.xlarge, c5a.2xlarge, c5a.4xlarge, c5a.8xlarge, c5a.12xlarge, c5a.16xlarge, c5a.24xlarge
- m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
- r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge, r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge

Ulteriori informazioni:

- Blog sulla tecnologia dei giochi di Amazon
- Amazon GameLift Serversprezzi delle istanze
- EC2 Istanze Amazon con processori AMD EPYC
- Forum Amazon GameLift Servers

11 novembre 2020: aggiornamento della versione all'SDK del server Amazon GameLift Servers

Versioni SDK aggiornate: Amazon GameLift Servers Server SDK 4.0.2

La nuova versione 4.0.2 di Server SDK corregge un problema noto relativo al funzionamento dell'API. `StartMatchBackfill()` Questa operazione restituisce ora una risposta corretta a una richiesta di match backfill.

Il problema non ha influito sul processo di match backfill e non è stata apportata alcuna modifica al funzionamento di questa funzionalità. Il problema potrebbe aver influito sulla messaggistica di registro e sulla gestione degli errori per le richieste di match backfill.

Scarica la versione più recente di Amazon GameLift Servers Server SDK subito Amazon GameLift Servers dopo.

5 novembre 2020: nuove personalizzazioni degli FlexMatch algoritmi

FlexMatchgli utenti possono ora modificare i seguenti comportamenti predefiniti per il processo di matchmaking. Queste personalizzazioni sono impostate in un set di regole di matchmaking. Non sono state apportate modifiche a. Amazon GameLift Servers SDKs

- Dai priorità ai ticket di backfill: Puoi scegliere di aumentare o diminuire la priorità dei ticket Match Backfill durante la ricerca di corrispondenze accettabili. Dare priorità ai ticket di riempimento

è utile quando è abilitata la funzione di riempimento automatico. Usa la proprietà algorithm. `backfillPriority`

- Preordina per ottimizzare la coerenza e l'efficienza delle partite: configura il tuo matchmaker in modo che preordini il pool di biglietti prima di raggruppare i ticket per la valutazione. Preordinando i ticket in base agli attributi chiave dei giocatori, le partite risultanti tendono ad avere giocatori più simili in questi attributi. Puoi anche aumentare l'efficienza del processo di valutazione preordinando gli stessi attributi utilizzati nelle regole delle partite. Utilizzate la proprietà dell'algoritmo `sortByAttributes` con la `strategy` proprietà impostata su «ordinato».
- Modifica il modo in cui vengono attivati i tempi di attesa per l'espansione: scegli se attivare le espansioni in base alla data del ticket più recente (impostazione predefinita) o più vecchio in una partita incompleta. L'attivazione sul ticket più vecchio tende a completare le partite più velocemente, mentre l'attivazione sul ticket più recente porta a una migliore qualità della partita. Usa la proprietà algorithm. `expansionAgeSelection`

17 settembre 2020: Amazon GameLift Servers aggiorna il server SDK

Versioni SDK aggiornate: Amazon GameLift Servers Server SDK 4.0.1

Il nuovo Server SDK contiene i seguenti aggiornamenti:

- API C# versione 4.0.1
  - L'operazione API non TerminateGameSession() è più supportata. Sostituiscila con una chiamata ProcessEnding() per terminare sia una sessione di gioco che il processo del server.
  - È stato risolto un problema noto relativo all'operazioneGetInstanceCertificate().
  - L'operazione restituisce GetTerminationTime() ora un valore del tipo di dati AwsDateTimeOutcome.
- API C++ versione 3.4.1
  - L'operazione non TerminateGameSession() è più supportata. Sostituiscila con una chiamata per ProcessEnding() terminare sia una sessione di gioco che il processo del server.
- Plugin Unreal Engine versione 3.3.2
  - L'operazione non TerminateGameSession() è più supportata. Sostituiscila con una chiamata per ProcessEnding() terminare sia una sessione di gioco che il processo del server.
  - L'operazione di callback `OnUpdateGameSession` viene aggiunta a per FProcessParametri supportare il match backfill.

Scarica la versione più recente di Amazon GameLift Servers Server SDK all'inizioAmazon GameLift Servers.

27 agosto 2020: Amazon GameLift ServersFleetIQ per l'hosting di giochi con Amazon EC2 (disponibilità generale)

Versioni SDK aggiornate: AWS SDK 1.8.36

La Amazon GameLift Servers FleetIQ soluzione per l'hosting di giochi a basso costo e basato sul cloud su Amazon EC2 è ora disponibile a tutti. Amazon GameLift ServersFleetIQoffre agli sviluppatori la possibilità di ospitare server di gioco direttamente su istanze Amazon EC2 Spot ottimizzando la loro fattibilità per l'hosting di giochi. Gli sviluppatori di giochi possono utilizzarli Amazon GameLift Servers FleetIQ con nuovi giochi o integrare la capacità dei giochi esistenti. Questa soluzione supporta l'uso di contenitori o altri AWS servizi come AWS Shield e Amazon Elastic Container Service (Amazon ECS).

Questa versione di disponibilità generale include i seguenti aggiornamenti della Amazon GameLift Servers FleetIQ soluzione:

- La nuova operazione API `DescribeGameServerInstances` restituisce informazioni, incluso lo stato, su tutte le istanze attive per un gruppo di server di Amazon GameLift Servers FleetIQ gioco.
- Nuova strategia di bilanciamento`ON_DEMAND_ONLY`, configura un gruppo di server di gioco per utilizzare solo istanze On-Demand. Puoi aggiornare la strategia di bilanciamento di un gruppo di server di gioco in qualsiasi momento, rendendo possibile passare dall'utilizzo delle istanze Spot alle istanze On-Demand secondo necessità.
- I seguenti elementi di anteprima sono stati eliminati per motivi di disponibilità generale:
  - Utilizzo di tasti di ordinamento personalizzati per le risorse del server di gioco. I server di gioco possono essere ordinati in base al timestamp di registrazione.
  - Etichettatura per le risorse del server di gioco.

16 aprile 2020: Amazon GameLift Servers aggiorna il server SDK per Unity e Unreal Engine

Versioni SDK aggiornate: Amazon GameLift Servers Server SDK 4.0.0, Local 1.0.5 Amazon GameLift Servers

La versione più recente dell'SDK del server Amazon GameLift Servers contiene i seguenti componenti aggiornati:

- C# SDK versione 4.0.0 aggiornata per Unity 2019.
- La versione 3.3.1 del plugin Unreal è stata aggiornata per le versioni 4.22, 4.23 e 4.24 di Unreal Engine.
- Amazon GameLift ServersVersione locale 1.0.5 aggiornata per testare le integrazioni che utilizzano la versione 4.0.0 dell'SDK del server C#.

Scarica la versione più recente di Server SDK subito dopo. Amazon GameLift ServersAmazon GameLift Servers

2 aprile 2020: Amazon GameLift ServersFleetIQ disponibile per l'hosting di giochi su EC2 (anteprima pubblica)

Versioni SDK aggiornate: AWS SDK 1.7.310

La caratteristica FleetIQ di Amazon GameLift Servers ottimizza l'uso delle istanze Spot a basso costo per l'hosting di giochi. Questa caratteristica è ora estesa per i clienti che desiderano gestire le proprie risorse di hosting direttamente anziché tramite il servizio Amazon GameLift Servers gestito. Questa soluzione supporta l'uso di contenitori o altri AWS servizi come AWS Shield e Amazon Elastic Container Service (Amazon ECS).

Ulteriori informazioni:

GameTech post sul blog su Amazon GameLift Servers FleetIQ

19 dicembre 2019: migliore AWS gestione delle Amazon GameLift Servers risorse

Versioni SDK aggiornate: AWS SDK 1.7.249

Ora puoi sfruttare gli strumenti di gestione delle AWS risorse con le risorse. Amazon GameLift Servers In particolare, a tutte le Amazon GameLift Servers risorse chiave (build, script, flotte, code di sessioni di gioco, configurazioni di matchmaking e set di regole di matchmaking) vengono ora assegnati valori Amazon Resource Name (ARN). Un ARN di risorse fornisce un identificatore coerente che è unico in tutte le regioni. AWS Possono essere utilizzati per creare politiche di autorizzazione specifiche per le risorse (AWS Identity and Access Management IAM). Alle risorse viene ora assegnato un ARN e anche l'identificatore di risorsa preesistente, che non è specifico della regione.

Inoltre, ora le risorse Amazon GameLift Servers supportano l'applicazione di tag. Puoi utilizzare i tag per organizzare le risorse, creare politiche di autorizzazione IAM per gestire l'accesso a gruppi di

risorse, personalizzare la ripartizione dei costi, ecc. AWS Quando si gestiscono i tag per le risorse Amazon GameLift Servers, utilizzare le operazioni API Amazon GameLift Servers, `TagResource()`, `UntagResource()` e `ListTagsForResource()`.

Ulteriori informazioni:

- TagResource nel Documentazione di riferimento API di Amazon GameLift Servers
- Assegnazione di tag alle risorse AWS nella Documentazione di riferimento generale AWS
- I nomi delle risorse Amazon nella AWS Guida generale

14 novembre 2019: nuovi AWS CloudFormation modelli, aggiornamenti nella regione Cina (Pechino)

Versioni SDK aggiornate: AWS SDK 1.7.210

AWS CloudFormation modelli per Amazon GameLift Servers

Amazon GameLift Serversle risorse possono ora essere create e gestite tramite AWS CloudFormation. I modelli di AWS CloudFormation build e fleet esistenti sono stati aggiornati per allinearli alle risorse attuali e ora sono disponibili nuovi modelli per script, code, configurazioni di matchmaking e set di regole di matchmaking. AWS CloudFormation i modelli semplificano notevolmente il compito di gestire gruppi di AWS risorse correlate, in particolare quando si distribuiscono giochi in più regioni.

Ulteriori informazioni:

- Amazon GameLift Serversriferimento al tipo di risorsa nella Guida per l'AWS CloudFormation utente
- Gestione di Amazon GameLift Servers hosting di risorse utilizzando AWS CloudFormationnella Guida per gli Amazon GameLift Servers sviluppatori

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.